AD-A038 399 CRAMER (H E) CO INC SALT LAKE CITY UTAH F/G 4/2
MIXING-LAYER ANALYSIS ROUTINE AND TRANSPORT/DIFFUSION APPLICATI--ETC(U)
MAR 77 R K DUMBAULD, J R BJORKLUND DAAD07-76-C-0023
UNCLASSIFIED TR-76-106-01 ECOM-77-2 NL

1 OF 3
AD
A038399

(12)

AD

Reports Control Symbol
OSD-1366

ADA038399

RESEARCH AND DEVELOPMENT TECHNICAL REPORT
ECOM-77-2

MIXING-LAYER ANALYSIS ROUTINE AND TRANSPORT/ DIFFUSION APPLICATION ROUTINE FOR EPAMS

By
R.K. Dumbauld
J.R. Bjorklund
H.E. Cramer Company, Inc.
University of Utah Research Park
P.O. Box 8049
Salt Lake City, Utah 84018

Contract Monitor: William Ohmstede

Atmospheric Sciences Laboratory
US Army Electronics Command
White Sands Missile Range, New Mexico 88002

March 1977

Approved for public release; distribution unlimited.

ECOM

UNITED STATES ARMY ELECTRONICS COMMAND - FORT MONMOUTH, NEW JERSEY 07703

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destory this report when it is no longer needed. Do not return it to the originator.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ECOM-77-2 / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): MIXING-LAYER ANALYSIS ROUTINE AND TRANSPORT/DIFFUSION APPLICATION ROUTINE FOR EPAMS.	5. TYPE OF REPORT & PERIOD COVERED: Final Report.
	6. PERFORMING ORG. REPORT NUMBER: TR-76-106-01
7. AUTHOR(s): R. K. Dumbauld and J. R. Bjorklund	8. CONTRACT OR GRANT NUMBER(s): DAAD07-76-C-0023
9. PERFORMING ORGANIZATION NAME AND ADDRESS: H. E. Cramer Company, Inc. University of Utah Research Park P.O. Box 8049 Salt Lake City, Utah 84108	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: DA Task No. 1S161102B53A
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Electronics Command Fort Monmouth, New Jersey 07703	12. REPORT DATE: March 1977
	13. NUMBER OF PAGES: 283
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Atmospheric Sciences Laboratory US Army Electronics Command White Sands Missile Range, New Mexico 88002	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

12 285p.

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Air pollution
Boundary layer modeling
Numerical analysis
Mathematical models
Atmospheric models
Sensitivity
Mesoscale prediction models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the development of two subelements of the U. S. Army Experimental Prototype Automatic Meteorological System (EPAMS). One of the subelements is a diagnostic routine (computer program) for the automatic calculation of the mixing depth, as well as the wind velocity and other meteorological parameters in the surface mixing layer, for a mesoscale calculation grid in complex terrain. The output from this routine is used with the second subelement--a transport/diffusion application routine--to calculate concentration/dosage fields

→ next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408235

20. Abstract (cont)

produced by pollutant emissions from selected sources located within the calculation grid. A two-layer shallow-fluid numerical model is used in the diagnostic routine to calculate the mesoscale wind field. The concentration/dosage fields calculated by the transport/diffusion application routine are obtained by solving generalized dispersion models for both continuous and instantaneous sources. In addition to descriptions of the modeling techniques, calculation procedures and computer program operations, example solutions for several case studies at White Sands Missile Range are also presented.

FOREWORD

This final report is submitted to the Atmospheric Sciences Laboratory, White Sands Missile Range in fulfillment of requirements under Contract No. DAAD07-76-C-0023.

The H. E. Cramer Company, Inc. is indebted to Mr. William Ohmstede, Atmospheric Sciences Laboratory, for the technical guidance and useful suggestions he provided in the development of the computer program routines described in this report.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

SECTION 1
INTRODUCTION

1.1 BACKGROUND

The Experimental Prototype Automatic Meteorological System (EPAMS) is part of an overall development program of the U. S. Army to provide a centralized data system for use in supporting future operations of the Field Army. Under the EPAMS development program, the Atmospheric Sciences Laboratory, White Sands Missile Range (WSMR) required implementation of an automated procedure for estimating the cloud trajectory and dispersion characteristics downwind from sources of contaminate material released at any arbitrary point within the atmospheric mixing layer. WSMR wished to implement these automated procedures using available technical approaches to prevent undue delays in the EPAMS development schedule.

The technical problem of estimating wind fields over complex terrain has been under continued investigation at WSMR. Because of the computer space/time criteria established for EPAMS, the model selected for use in EPAMS had to be relatively simple in concept. The computer algorithms developed by Tingle and Bjorklund (1973) for WSMR's use in predicting wind fields in complex terrain were ideally suited for this purpose. The algorithms, based on the shallow-fluid equations of oceanography, have simple input requirements, reach stable solutions in a relatively few iterations and thus conserve computation time, and have been shown to produce wind field estimates agreeing with observed estimates in complex terrain surrounding WSMR and Dugway Proving Ground, Utah. Similar criteria applied to the selection of dispersion models for use in EPAMS. The generalized dispersion models developed under previous Government contracts for predicting concentration, dosage and deposition downwind from nearly-instantaneous and continuous volume sources (Cramer, et al., 1972; Dumbauld, et al., 1970, 1973) had been validated for use in conjunction with military activities. These computerized models were

developed with the intention of providing the U. S. Army with a universally adaptable means of calculating dispersion characteristics for all types of terrain, meteorological conditions and types of sources.

Thus, the technical data base requisite for development of an automated procedure for predicting cloud trajectories and dispersion characteristics in complex terrain existed. The principle technical objectives of the work described below were to utilize the existing technical data base in the:

- Development of a diagnostic/prognostic analysis routine for evaluating the depth and wind velocity of the atmospheric mixing layer in association with complex terrain

- Development of a diagnostic/prognostic application routine for evaluating the transport and diffusion of contaminant material released from a source within the mixing layer

and incorporation of these routines into EPAMS as subordinate elements.

Figure 1-1 is a simplified schematic diagram of the EPAMS functional elements illustrating the interface of the Analysis and Application Routines or Phases with EPAMS. The Applications Phase is called by the EPAMS Executive as a result of a request for information routed through a Message Management Routine. The Applications Phase extracts the necessary meteorological and source information from the Data Base Management Routine and calculates the transport and diffusion of contaminant material released to the atmosphere. The solution is output to the Data Base, where it is made available to the EPAMS Executive for forming a response to the information request. The Analysis Phase is called by the EPAMS Executive as a result of a standing request for a routine analysis of meteorological conditions in the

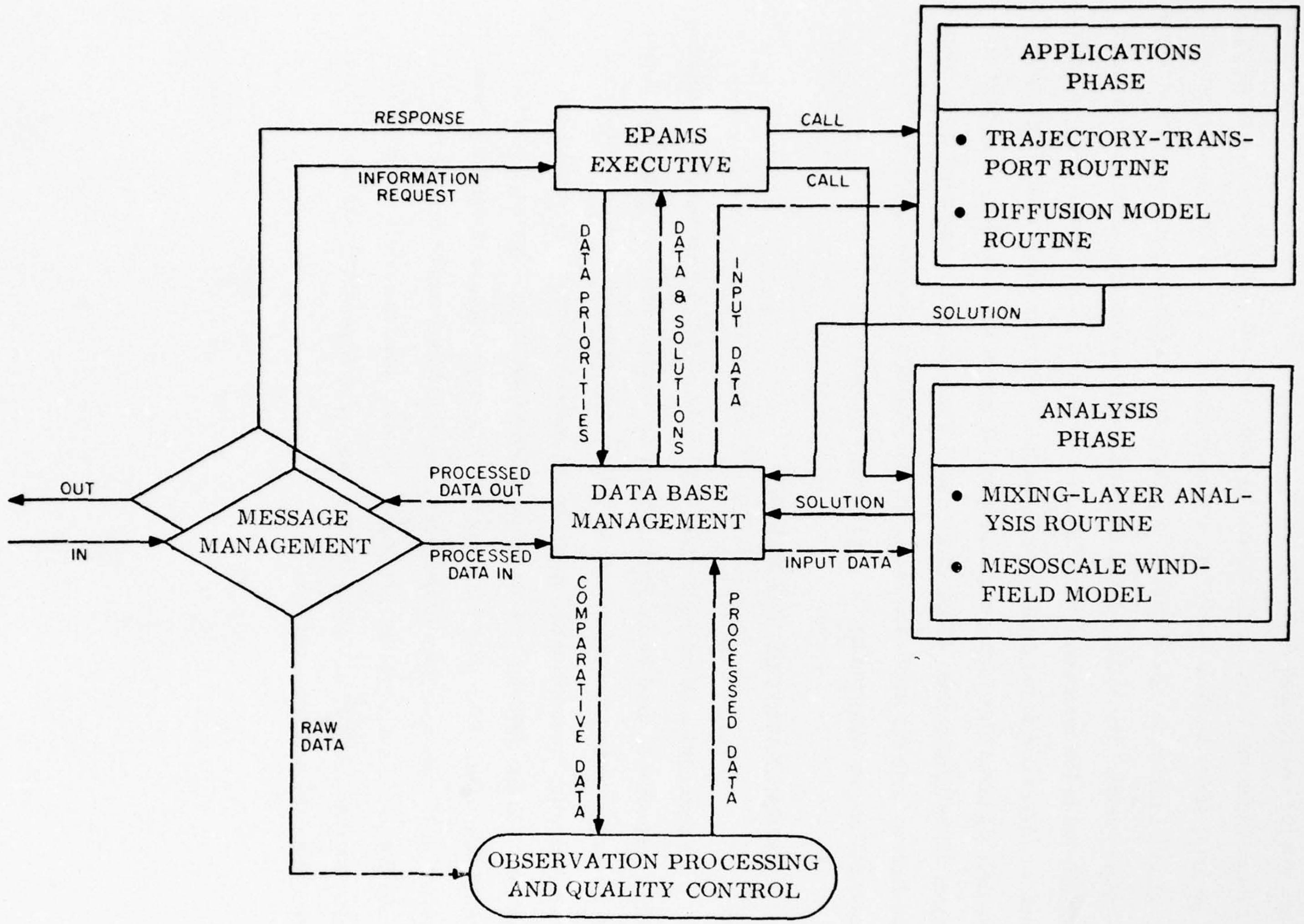


Figure 1-1. Schematic Diagram of the EPAMS Functional Elements.

surface mixing layer. The Analysis Phase extracts upper-air and surface meteorological data, including U. S. A. F. Global Weather Central (GWC) "fine mesh" prediction data, from the Data Base. As shown in Figure 1-1, the Analysis Phase is comprised of two major subelements, a mixing-layer analysis routine and mesoscale wind-field model. The mixing-layer analysis routine calculates initialization parameters for use in the mesoscale wind-field model as well as additional meteorological inputs for direct use in the Applications Routine. The mesoscale model calculates wind-field patterns and mixing-depth heights for use in the trajectory-transport subelement of the Applications Phase of EPAMS. The Analysis Phase does not communicate directly with the Applications Phase, but supplies solutions to the data base for subsequent use as required by the Applications Phase.

1.2 ORGANIZATION OF THE REPORT

A description of the mesoscale wind-field model and the transport diffusion models respectively used in the Analysis and Application Phases is provided in Section 2 below. Section 3 contains a description of procedures used in the Analysis Phase to develop meteorological inputs to the mesoscale wind-field model and for direct input to the diffusion models from meteorological data extracted from the EPAMS data base. Details of the computer program structure of theAnalysis and Application Phases are given in Section 4. Section 5 contains example cases illustrating the use of the computer program. Detailed user instructions for the computer program routines are given in Appendix A and Appendix B contains a listing of the computer program routines.

SECTION 2

MESOSCALE WIND-FIELD AND TRANSPORT/DIFFUSION MODELS

2.1 MESOSCALE WIND-FIELD MODEL

The Analysis Phase developed for EPAMS relies on a numerical modeling technique developed for WSMR by Tingle and Bjorklund (1973) to calculate details of the mesoscale wind-field and depth of the surface-mixing layer over complex terrain. The computer algorithm, based on the shallow-fluid equations of oceanography, describes the motion of fluid in two dimensions by the expressions:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g'\frac{\partial}{\partial x}\left(H_m + z_T\right) = 0 \qquad (2\text{-}1)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g'\frac{\partial}{\partial y}\left(H_m + z_T\right) = 0 \qquad (2\text{-}2)$$

$$\frac{\partial H_m}{\partial t} + \frac{\partial\left(uH_m\right)}{\partial x} + \frac{\partial\left(vH_m\right)}{\partial y} = 0 \qquad (2\text{-}3)$$

where

u, v = x, y components of the fluid velocity in the layer next to the terrain

H_m = depth of the surface mixing-layer

z_T = height of the terrain

g' = reduced acceleration of gravity

$$= g\left(1 - \frac{\rho_1}{\rho_o}\right)$$

g = acceleration of gravity

ρ_o = density of fluid in the surface mixing layer

ρ_1 = density of fluid in the deep passive layer above the fluid in the surface mixing-layer

Thus the equations can be used to describe mesoscale wind-patterns over complex terrain when the atmosphere above the terrain can be distinctively divided into at least two layers of air of different densities. Such an atmospheric condition arises when the mixing layer above the terrain is capped by an elevated inversion. The warmer upper layer interacts with the cooler lower layer only in that its lesser density reduces the speed of gravity waves introduced into the lower layer. In the algorithm, Tingle and Bjorklund have written Equations (2-1), (2-2) and (2-3) in their momentum form. Mesoscale wind patterns are generated by impulsively accelerating the momentum in the lower layer to a constant value and, after a empirically determined model time (see Section 4), the flow in the region of interest is assumed to be stabilized (steady state).

The computer algorithm utilizes an explicit method of the Lax-Wendroff type of finite differencing technique using nine grid points and one time level in combination with a nine-point low-pass filter. The differencing scheme is designed such that the finite differencing forms are centered to preserve high-order accuracy. The grid arrangement for this centering scheme and finite differencing form of the equations are fully described in the report by Tingle and Bjorklund (1973). As noted above, high-frequency oscillations generated in critical flow situations near regions where hydraulic jumps occur are removed by a filtering technique. The filtering technique used in the algorithm is based on a suggestion from Shapiro (1970) and has the advantage that its smoothing effect is easily determined from its response

function. In the Analysis Phase, the filter is applied every seventh time step. It is well known that the length of the time step Δt used in obtaining the solution also effects the stability of the finite difference solution. The time step is altered automatically in this algorithm in accordance with the expression

$$\Delta t = \frac{\lambda}{\sqrt{2}} \frac{\Delta x_{min}}{\left[\sqrt{u^2 + v^2} + \sqrt{gH_m}\right]_{max}} \tag{2-4}$$

where the stability parameter λ is dependent on the grid spacing Δx and varies between zero and one. For the present application using a minimum spacing (Δx_{min}) of 5 kilometers, the stability parameter has been set equal to 0.4. Because the entire solution grid must be searched to determine the maximum wave speed in the denominator of Equation (2-4), the value of Δt is calculated every second time step rather than every time step to save computation time.

A sensitivity analysis of the two-layer mesoscale wind field model currently in progress under separate contract with U. S. Army Dugway Proving Ground has indicated model instability can frequently occur in localized terrain regions where the inversion height H_m is initialized, or becomes adjusted by the model, to altitudes within a few tens of meters of the higher terrain elevations. The momentum conservation properties of the computational scheme are such that wind velocities in these localized regions may become unreasonably large as the layer thickness approaches a minimum value. In order to prevent this type of solution instability, the computer algorithm has been redesigned so that the u and v components of the solution at any given time step are automatically set to zero and the thickness raised to a minimum depth of 30 meters if the calculated depth for a grid point is less than 30 meters. Limited experience in the application of this feature

of the mesoscale model indicates that computational stability is preserved without undue loss of momentum over the entire grid and with the result that wind fields in the immediate vicinity of higher elevations under these conditions appear to be reasonable (see Example in Section 5).

The computerized algorithm uses an expanding grid in the boundary area beyond the terrain region of interest to damp outgoing waves and eliminate incoming waves based on the technique suggested by Lavoie (1972). Grid points in the boundary region are spaced at 10, 20, 40, 80 and 160 kilometers from all four sides of the interior grid. The terrain height at all points in the boundary region are set equal to the minimum elevation within the interior grid. Since the shallow-water equations are written in their momentum form, the initial momentum is set equal to the momentum used in initializing parameters in the interior grid.

Readers interested in the form of the finite-differenced equations and other details on which the mesoscale wind-field computer algorithm is based are referred to the paper by Tingle and Bjorklund (1973).

2.2 TRANSPORT-DIFFUSION MODELS

2.2.1 Trajectory/Transport Routine

The Trajectory-Transport Routine incorporated in the Applications Phase of EPAMS shown in Figure 1-1 is designed to calculate cloud trajectories and mean-layer winds and mixing-layer depths at fixed intervals along the trajectories using the output from the mesoscale wind-field model described above. The trajectory of a cloud released at any point within the model solution grid is calculated using a simple Euler predictor-corrector scheme and nine-point interpolation procedure. The computer program uses wind vector data from the mesoscale wind-field model solution at nine grid points surrounding the source (point of release) and,

through the interpolation procedure, estimates the u and v components of the wind at the point of release. The first approximation to the first point on the trajectory curve is then determined by resolving the components and placing a point at a fixed interval along the resultant vector. The fixed distance interval is set at one-tenth the grid spacing used in the mesoscale wind-field solution matrix so that in the present program a fixed interval of 500 meters is used. Wind velocity components at this point are again estimated from the nine-point interpolation procedure. These new u and v components are then averaged with the previously determined components at the starting point. The final estimate of the first point on the trajectory is then calculated by resolving the averaged components and moving a fixed interval along the resultant vector. This final estimate of the first point on the trajectory is then treated in the same manner as the initial starting point (source) and the interpolation and iteration scheme repeated to find the second point on the trajectory. This procedure is repeated until the trajectory passes outside the grid area containing wind-vector data or the number of iterations exceeds a predetermined limit.

The interpolation method is based on an evaluation of a truncated two-dimensional Taylor's expansion formula (Carnahan, Luther and Wilkes, 1969, p. 430) given by

$$f(x, y) \cong f_{i,j} + hf_x + kf_y + \frac{h^2}{2} f_{xx} + hkf_{xy} + \frac{k^2}{2} f_{yy} \qquad (2\text{-}4)$$

where

$h = x - x_i$

$k = y - y_j$

and $f_{i,j}$ is either the u or v component at x_i, y_j, the nearest grid point to the point of interpolation (x, y). The partial derivatives f_x, f_y, f_{xx}, f_{xy} and f_{yy} are estimated by divided differences.

The depth of the mixing layer at fixed intervals along the trajectory is also interpolated from data provided by the mesoscale wind-field model using a four-point bivariate interpolation formula given by

$$H_m(x,y) = (1-p)(1-q) H_{m_{i,j}} + p(1-q) H_{m_{i+1,j}} + q(1-p) H_{m_{i,j+1}} + pq H_{m_{i+1,j+1}} \quad (2\text{-}5)$$

where

$$p = (x - x_i)/(x_{i+1} - x_i)$$

$$q = (y - y_j)/(y_{j+1} - y_j)$$

and $H_{m_{i,j}}$, $H_{m_{i+1,j}}$, $H_{m_{i,j+1}}$, and $H_{m_{i+1,j+1}}$ are the mixing layer depths at the four surrounding grid points.

2.2.2 Diffusion Models Routine

The Diffusion Models Routine shown in Figure 1-1 as an element of the Applications Phase of EPAMS is comprised of volume source diffusion models for instantaneous and continuous sources and models for use in predicting the buoyant rise of plumes from continuous sources. These models are used in the Diffusion Models Routine to estimate isopleths of dosage and concentration, with or without

depletion due to simple decay, gravitational settling and precipitation scavenging, along the cloud trajectory determined by application of the Trajectory-Transport Routine.

Instantaneous Volume-Source Model

The volume-source model for use in EPAMS is similar to the models for instantaneous sources described by Cramer, et al. (1972) and by Dumbauld, Bjorklund and Bowers (1973). The concentration for an instantaneous volume-source can be expressed as the product of four terms:

$$\chi_n = \{\text{Peak Term}\}\{\text{Lateral Term}\}\{\text{Vertical Term}\}\{\text{Depletion Term}\} \quad (2\text{-}6)$$

The Peak Term describes the concentration at the cloud centerline and is given by the expression

$$\text{Peak Term} = \frac{Q}{(2\pi)^{3/2}\,\sigma_{xn}\,\sigma_{yn}\,\sigma_{zn}} \quad (2\text{-}7)$$

where

Q = total source strength

σ_{xn} = standard deviation of the concentration distribution in the alongwind direction at the n^{th} distance along the cloud trajectory

σ_{yn} = standard deviation of the concentration distribution in the crosswind direction at the n^{th} distance along the cloud trajectory

σ_{zn} = standard deviation of the concentration distribution in the vertical at the n^{th} distance along the cloud trajectory

The Vertical Term describes the vertical distribution of material at the n^{th} distance along the cloud trajectory as modified by reflections of material at the top of the surface mixing layer and at the earth's surface and by gravitational settling of particulates

$$
\begin{aligned}
\text{Vertical Term} = \sum_{a=0}^{\infty} \Bigg\{ & r^{a} \exp\left[-\frac{1}{2}\left(\frac{2a\,H_{m,n} - H' + z + V_s \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}}{\sigma_{z,n}}\right)^2\right] \\
& + r^{a+1} \exp\left[-\frac{1}{2}\left(\frac{2a\,H_{m,n} + H' + z - V_s \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}}{\sigma_{z,n}}\right)^2\right] \Bigg\} \\
+ \sum_{a=1}^{\infty} \Bigg\{ & r^{a} \exp\left[-\frac{1}{2}\left(\frac{2a\,H_{m,n} + H' - z - V_s \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}}{\sigma_{z,n}}\right)^2\right] \\
& + r^{a-1} \exp\left[-\frac{1}{2}\left(\frac{2a\,H_{m,n} - H' - z + V_s \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}}{\sigma_{z,n}}\right)^2\right] \Bigg\}
\end{aligned}
\tag{2-8}
$$

where, for convenience in writing the Vertical Term, 0^0 (zero to the zeroeth power) is set equal to unity and

H' = effective release height (see Equation (2-20))

z = calculation height

$H_{m,n}$ = mixing layer depth from mesoscale model for n^{th} distance along the trajectory

r = fraction of material reflected at the surface (1 for complete reflection and 0 for no reflection)

V_s = settling velocity for material in a given size category

Δx_n = incremental distance along the cloud trajectory

$\bar{u}_n$ = mean wind speed in the layer containing the cloud at the n^{th} point along the trajectory (see Equation (2-17) below)

The Lateral Term is given by

$$\text{Lateral Term} = \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_{yn}}\right)^2\right] \tag{2-9}$$

where

y = crosswind distance from the centerline of the cloud trajectory

The Depletion Term refers to the loss of material by simple decay processes and by precipitation scavenging. The form of the Depletion Term for each of these processes is:

$$\text{(Decay)} \qquad \exp\left[-k \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}\right] \tag{2-10}$$

$$\text{(Precipitation Scavenging)} \qquad \exp\left[-\Lambda \sum_{n=1}^{N} \frac{\Delta x_n}{\bar{u}_n}\right] \tag{2-11}$$

where

k = decay coefficient or fraction of material lost per unit time

Λ = washout coefficient or fraction of material removed by scavenging per unit time

The subset of equations describing the standard deviations of the alongwind σ_{xn}, crosswind σ_{yn} and vertical σ_{zn} concentration distributions is:

$$\sigma_{xn} = \frac{L\{x_n\}}{4.3} + \sigma_{x(n-1)} = \frac{0.6\,\Delta\bar{u}_n}{4.3\,\bar{u}_n}\,\Delta x_n + \sigma_{x(n-1)} \tag{2-12}$$

where

$L\{x_n\}$ = alongwind dimension of the cloud

$\Delta\bar{u}_n$ = wind speed shear at the n^{th} point along the trajectory

$$= \frac{\bar{u}_{5,n}}{5^{p_n}}\left(z_{T,n}^{p_n} - z_{B,n}^{p_n}\right) \tag{2-13}$$

$\bar{u}_{5,n}$ = wind speed at a height of five meters at the n^{th} point along the trajectory

p_n = wind power-law coefficient at the n^{th} point along the trajectory

$z_{T,n}$ = top of the cloud at the n^{th} point

$$= \begin{Bmatrix} H' + 2.15\,\sigma_{zn} \,; & z_{T,n} < H_{m,n} \\ H_{m,n} \,; & z_{T,n} \geq H_{mn} \end{Bmatrix} \tag{2-14}$$

$z_{B,n}$ = base of the cloud at the n^{th} point

$$= \begin{cases} H' - 2.15\,\sigma_{zn} \; ; \; z_{B,n} \geq 0 \\ 0 \; ; \; z_{B,n} \leq 0 \end{cases} \tag{2-15}$$

$$\bar{u}_{5,n} = \frac{\bar{\bar{u}}_n \left(H_{m,n} - 5\right) 5^{p_n} \left(1 + p_n\right)}{H_{m,n}^{(1 + p_n)} - 5^{(1 + p_n)}} \tag{2-16}$$

$\bar{\bar{u}}_n$ = mean layer wind speed from mesoscale model at n^{th} point

$\bar{u}_n$ = mean wind speed in the layer containing the cloud

$$= \frac{\bar{u}_{5,n} \left(z_{T,n}^{(1 + p_n)} - z_{B,n}^{(1 + p_n)}\right)}{(1 + p)\,(5)^{p_n} \left(z_{T,n} - z_{B,n}\right)} \tag{2-17}$$

Δx_n = incremental distance between the n points equal to 500 meters

$$\sigma_{yn} = \sigma'_{A,n} \left[\Delta x_n + \left(\frac{\sigma_{y,n-1}}{\sigma'_{A,n}}\right)^{1/\alpha}\right]^{\alpha} \tag{2-18}$$

where

$\sigma'_{A,n}$ = standard deviation of the wind aximuth angle at the n^{th} point

$$= \sigma_{A,n} \left(\frac{\pi}{180}\right) \left(\frac{H'}{5}\right)^{m} \left(\frac{\tau}{600}\right)^{0.2} \tag{2-19}$$

$$H' = \begin{cases} H \; ; \; H \geq 5 \\ 5 \; ; \; H < 5 \end{cases} \tag{2-20}$$

τ = source function time $\cong 2.5$ seconds for instantaneous sources

$\sigma'_{A,n}$ = ten-minute standard deviation of the wind azimuth angle

$$\sigma_{z,n} = \sigma'_{E,n} \left[\Delta x_n + \left(\frac{\sigma_{z,n-1}}{\sigma_{E,n}} \right)^{1/\beta} \right]^{\beta} \qquad (2-21)$$

where

$$\sigma'_{E,n} = \sigma_{E,n} (\pi/180) (H'/5)^n \qquad (2-22)$$

$\sigma_{E,n}$ = standard deviation of the wind elevation angle

The cloud expansion coefficients α (lateral) and β (vertical) in the above equations are set to unity for instantaneous sources.

The concentration χ_n is calculated under the additional restriction that $\chi_n \leq \chi_{n-1}$. If $\chi_n > \chi_{n-1}$, the values of $\sigma_{y,n}$ and $\sigma_{x,n}$ are increased proportionately so that $\chi_n = \chi_{n-1}$ and the calculations continued.

Dosage for an instantaneous source is obtained from the expression

$$D_n = \frac{\chi_n \sqrt{2\pi}\ \sigma_{xn}}{\bar{u}_{z,n}} \qquad (2-23)$$

where

$\bar{u}_{z,n}$ = mean wind speed at the calculation height z and n^{th} point on the trajectory

$$= \left\{ \begin{array}{ll} \bar{u}_{5,n}\left(\frac{z}{5}\right)^{p_n} & ; \; z > 1 \\ \bar{u}_{5,n}\left(\frac{1}{5}\right)^{p_n} & ; \; z \leq 1 \end{array} \right\} \tag{2-24}$$

Continuous Source Model

The continuous-source model for use in EPAMS is a modified version of the Gaussian model for continuous sources described by Pasquill (1962) in which the concentration is also expressed as the product of four terms:

$$\chi_{c,n} = \{\text{Peak Term}\}\{\text{Lateral Term}\}\{\text{Vertical Term}\}\{\text{Depletion Term}\} \tag{2-25}$$

The Peak Term of the continuous source concentration model is given by the expression

$$\text{Peak Term} = \frac{Q'}{2\pi\, \sigma_{zn}\, \sigma'_{yn}\, \bar{u}_n} \tag{2-26}$$

where

Q' = source strength expressed as a rate

σ'_{yn} = standard deviation of the concentration distribution in the crosswind direction for a continuous source at the n^{th} distance along the plume trajectory

and $\bar{u}_n$ and σ_{zn} are respectively defined by Equations (2-17) and (2-19). The Vertical, Depletion and Lateral Terms (with σ_{yn} replaced by σ'_{yn} where appropriate)

are identical to the similar terms described for the instantaneous volume source. The subset of equations describing σ'_{yn} is:

$$\sigma'_{yn} = \sigma'_{Ac,n}\left[\Delta x_n + \left(\frac{\sigma'_{y,n-1}}{\sigma'_{Ac,n}}\right)^{1/\alpha}\right]^{\alpha} \tag{2-27}$$

where

$$\sigma'_{Ac,n} = \sigma_{A,n}\left(\frac{\pi}{180}\right)\left(\frac{H'}{5}\right)^{m} \tag{2-28}$$

and the remaining parameters have previously defined. The default values of α and β used in the program for continuous sources are $\alpha = 0.9$ and $\beta = 1$.

Plume-Rise Models

The effective height H of a buoyant plume is given by the sum of the physical stack height h and the buoyant rise Δh. For a neutral or unstable atmosphere, the effective stack height is given by (after Briggs, 1971; 1972)

$$H = h + f\left\{\frac{15.98}{\bar{u}\{h\}}\left[r_s^2\, w\left(1 - \frac{T_s}{T_c}\right)\right]^{1/3} h^{2/3}\right\} \tag{2-29}$$

where

r_s = the inner radius of the stack at the exit (m)

w = the stack exit velocity $\left(m/sec^{-1}\right)$

T_c = the stack exit temperature (^{o}K) for the continuous source

T_s = ambient air temperature (^{o}K)

f = empirical correction factor

h = actual stack height

$\bar{u}\{h\}$ = mean wind speed at stack height

$$= \bar{u}_{5,\, n=1}\left(\frac{h}{5}\right)^{p_{n=1}} \qquad (2\text{-}30)$$

The factor f, which limits the plume rise as $\bar{u}\{h\}$ approaches the stack exit velocity w, is defined by

$$f = \begin{cases} 1 & ; \bar{u}\{h\} \leq w/1.5 \\ \left(\dfrac{3w - 3\bar{u}\{h\}}{w}\right) & ; w/1.5 < \bar{u}\{h\} < w \\ 0 & ; \bar{u}\{h\} \geq w \end{cases} \qquad (2\text{-}31)$$

Equation (2-29) is used in the EPAMS program when the net radiation index equals or exceeds 2, $\bar{u}\{h\}$ equals or exceeds 5 meters per second, or the potential temperature lapse rate $\bar{\Phi}$ is equal to or less than 0. When these conditions are not met, the program uses the corresponding Briggs (1971) rise formula for a stable atmosphere given by

$$
H = \begin{cases} h + 2.397\, f\left[\dfrac{wr_s^2\, T_s}{\bar{u}\{h\}\,\Phi}\left(1 - \dfrac{T_s}{T_c}\right)\right]^{1/3} ; & \dfrac{\pi\, u\{h\}}{10\, S^{1/2}} < h \\[2ex] h + 1.903\, f\left\{\dfrac{wr_s^2\, T_s}{\bar{u}\{h\}\,\Phi}\left[1 - \cos\left(\dfrac{10\, S^{1/2}\, h}{\bar{u}\{h\}}\right)\right]\right\}^{1/3} ; & \dfrac{\pi\, \bar{u}\{h\}}{10\, S^{1/2}} \geq h \end{cases} \tag{2-32}
$$

where

S = stability parameter

$$
= \frac{9.8}{T_s}\,\bar{\Phi} \tag{2-33}
$$

$\bar{\Phi}$ = height-weighted mean potential temperature lapse rate over an approximately 200-meter interval above h

$$
= \frac{\sum_{z \simeq h}^{z \simeq h+200} \Phi_z\, \Delta Z_z}{\sum_{z \simeq h}^{z \simeq h+200} \Delta Z_z} \tag{2-34}
$$

$$
\Phi_z = \frac{T_{A,z+1} - T_{A,z}}{\Delta Z_z} \tag{2-35}
$$

$$
\Delta Z_z = Z_{z+1} - Z_z \tag{2-36}
$$

$T_{A,z}$ = potential temperature at height Z_z

$T_{A,z+1}$ = potential temperature at height Z_{z+1}

SECTION 3

DATA ANALYSIS PROCEDURES

The EPAMS data base contains data blocks of rawinsonde, USAF GWC predicted wind and temperature data, and surface meteorological information. Section 3 describes the automated data analysis procedures required to utilize this data in the development of parametric inputs to the mesoscale wind-field and diffusion models. The Mixing-Layer Analysis Routine described in Section 3.1 below is designed to provide initialization parameters required by the mesoscale wind-field model. The Diffusion Model Routine, described in Section 3.2, uses surface meteorological data and mixing layer parameters provided by the mesoscale wind-field model as direct inputs to the diffusion model and for use in defining other model input parameters.

3.1 MIXING-LAYER ANALYSIS ROUTINE

The Mixing-Layer Analysis Routine uses a relatively simplistic approach to utilize the various types of data available for the EPAMS data base in the calculation of initialization parameters for the mesoscale wind-field model. The decision to maintain a simple approach in this initial effort to develop a comprehensive prediction program was made so that the effects of logic decisions on the derived parameters could be easily traced and analyzed. Based on this reasoning, the analysis routine is currently designed to perform a modified "single-station analysis" rather than a more complex analysis of the pressure, wind and temperature fields over the entire region of interest. It should also be noted that some of the logic decisions made in accepting, rejecting and analyzing the data are somewhat arbitrary at this point in the development of the program. Where time has permitted, we have attempted to study the effects of the various decision processes. However, until the program is used to predict dosage and concentration fields under a wide variety of meteorological conditions, the total implications of many of the decision processes will remain unknown.

Features of the Mixing-Layer Analysis Routine are portrayed in the schematic logic diagram shown in Figure 3-1. Inspection of Figure 3-1 shows that the key features of the procedure are:

- Rawinsonde data from the closest station with a radius of less than 100 kilometers from the release point and less than three-hours old at the time of request will be used without modification to obtain estimates of the input parameters to the mesoscale model

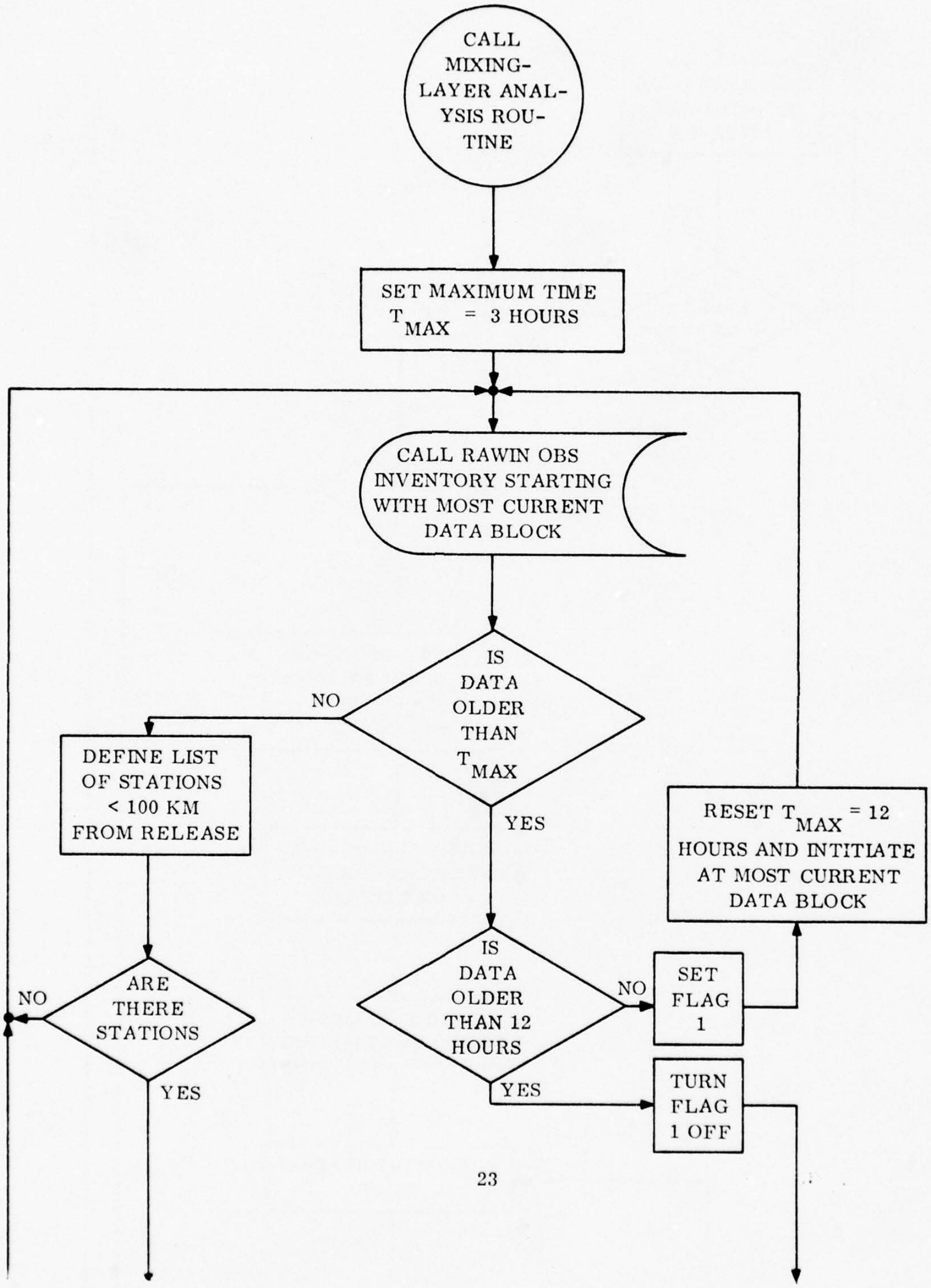
- Rawinsonde data measured at the closest station less than 100 kilometers from the release point, greater than three-hours old and less than twelve-hours old are subject to modification using GWC predicted and surface winds and temperatures

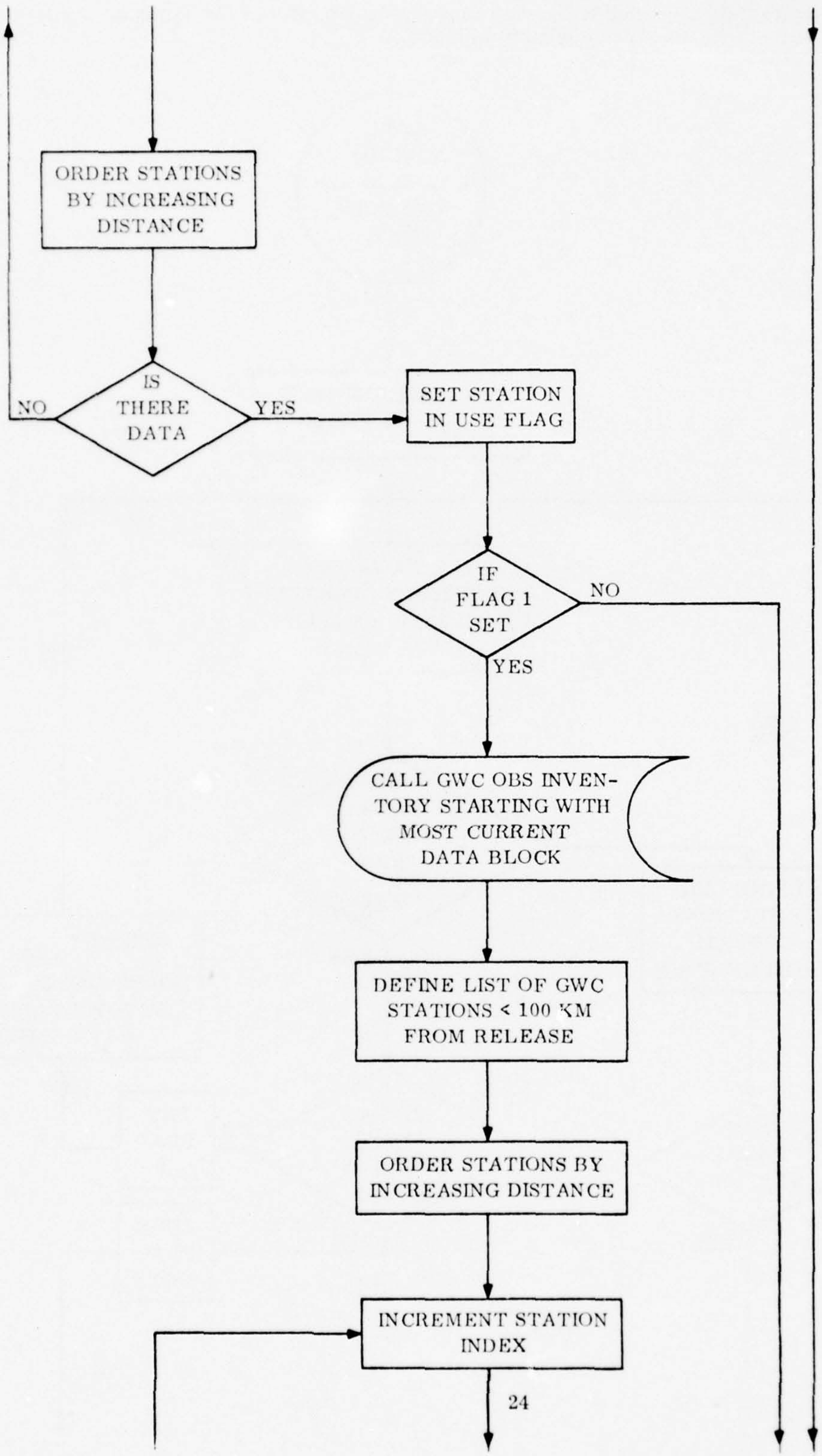
- Climatological estimates of mixing-layer depths and surface wind speeds are used as mesoscale model inputs if all rawinsonde data is more than twelve hours old or if there are no stations within a 100 kilometer radius of the release point

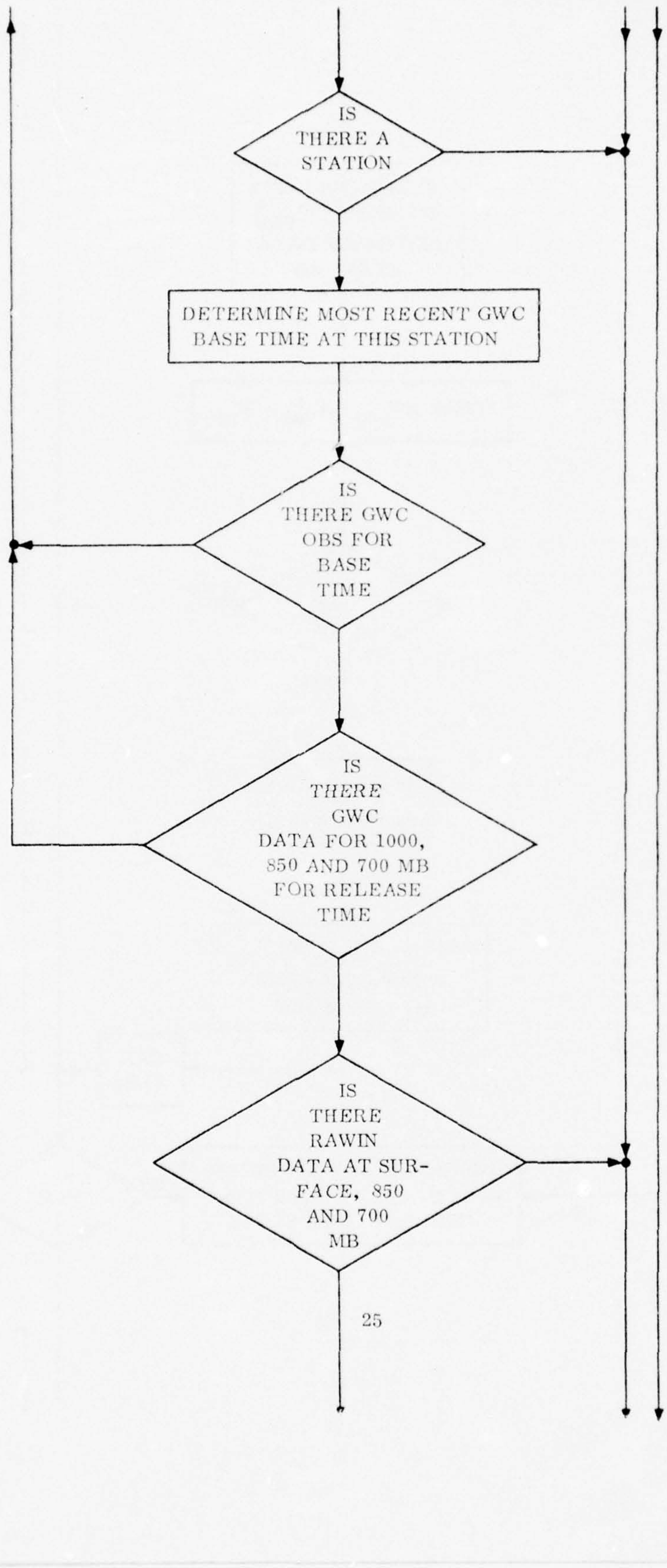
- Climatological estimates are also used if the rawinsonde data is greater than six-hours old and the rawinsonde surface temperature and 850 millibar temperatures have not been modified because of unavailable or inadequate GWC and surface data

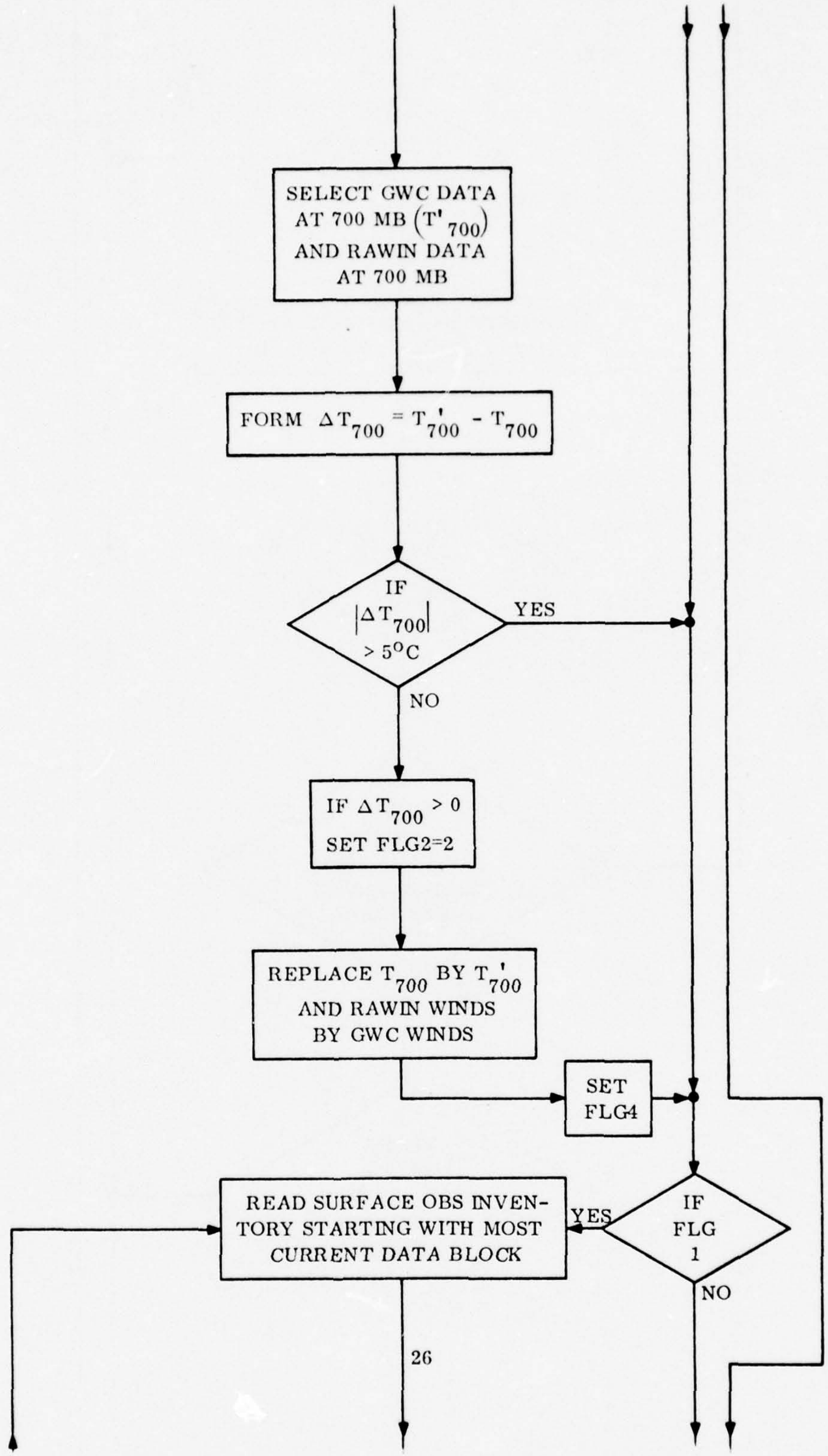
- GWC data is used to modify the rawinsonde temperatures and winds at the 700 and 850 millibar levels; rawinsonde

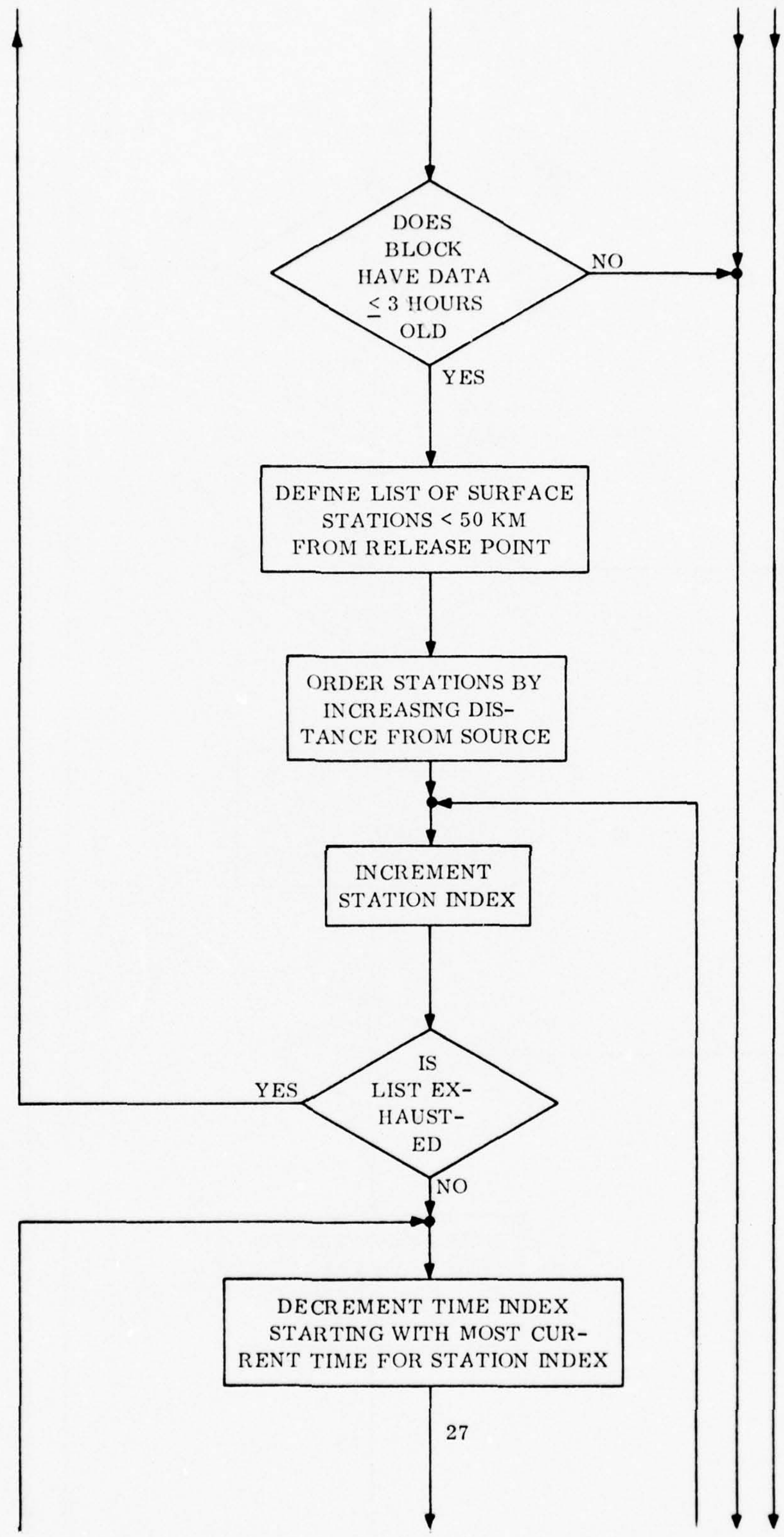
Figure 3-1. Schematic diagram illustrating procedures for obtaining inputs to the mesoscale wind field model.

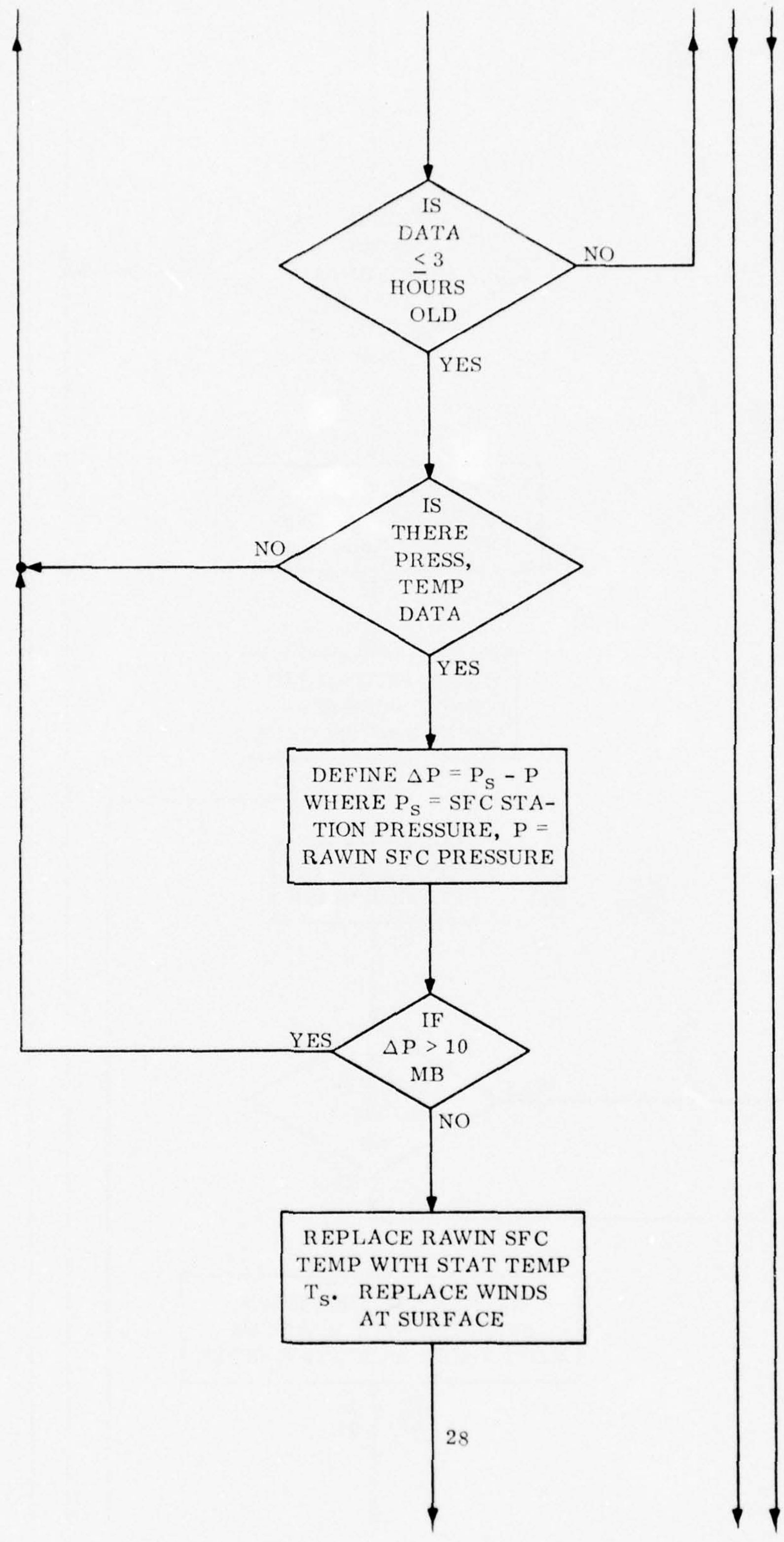


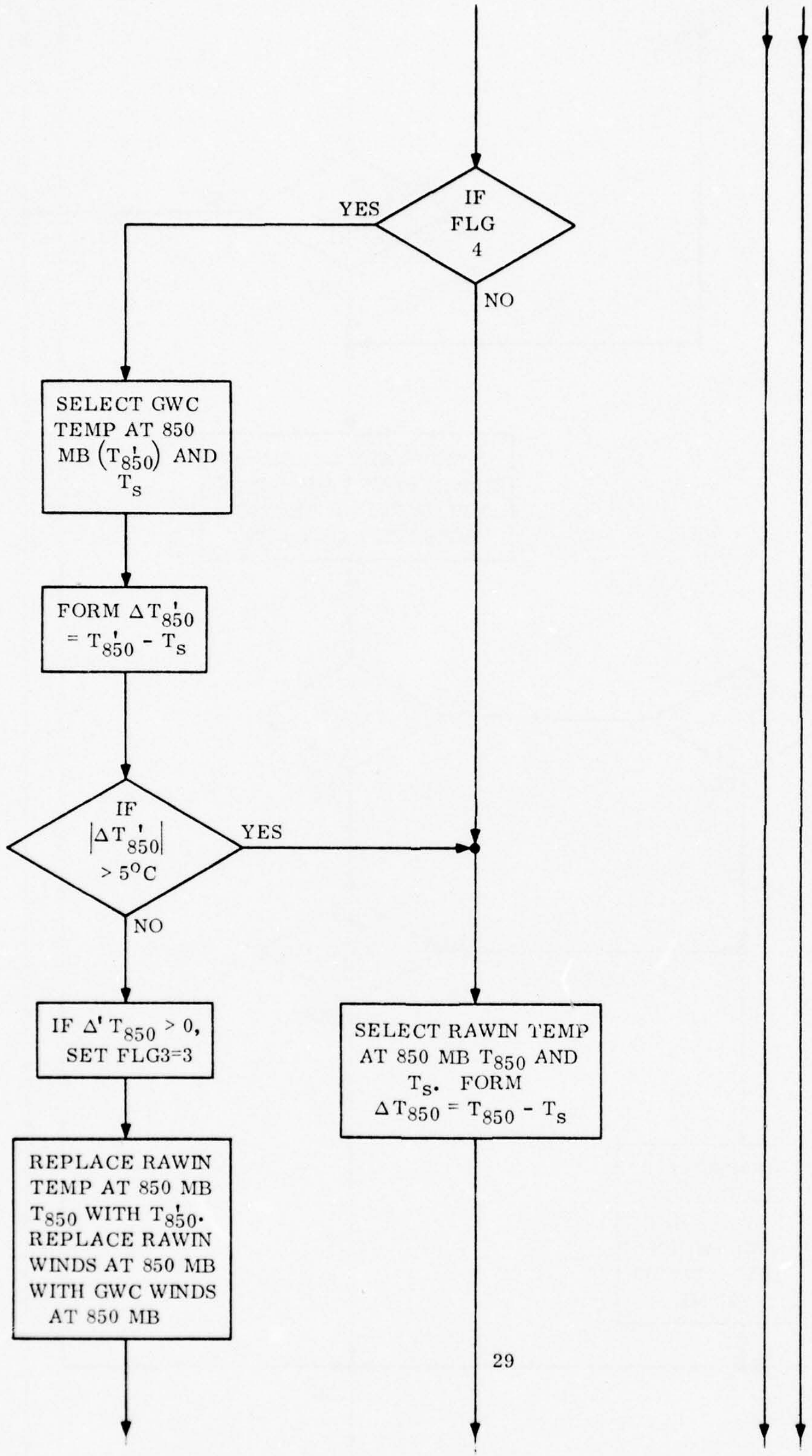

ORDER STATIONS BY INCREASING DISTANCE
IS THERE DATA
NO
YES
SET STATION IN USE FLAG
IF FLAG 1 SET
NO
YES
CALL GWC OBS INVENTORY STARTING WITH MOST CURRENT DATA BLOCK
DEFINE LIST OF GWC STATIONS < 100 KM FROM RELEASE
ORDER STATIONS BY INCREASING DISTANCE
INCREMENT STATION INDEX

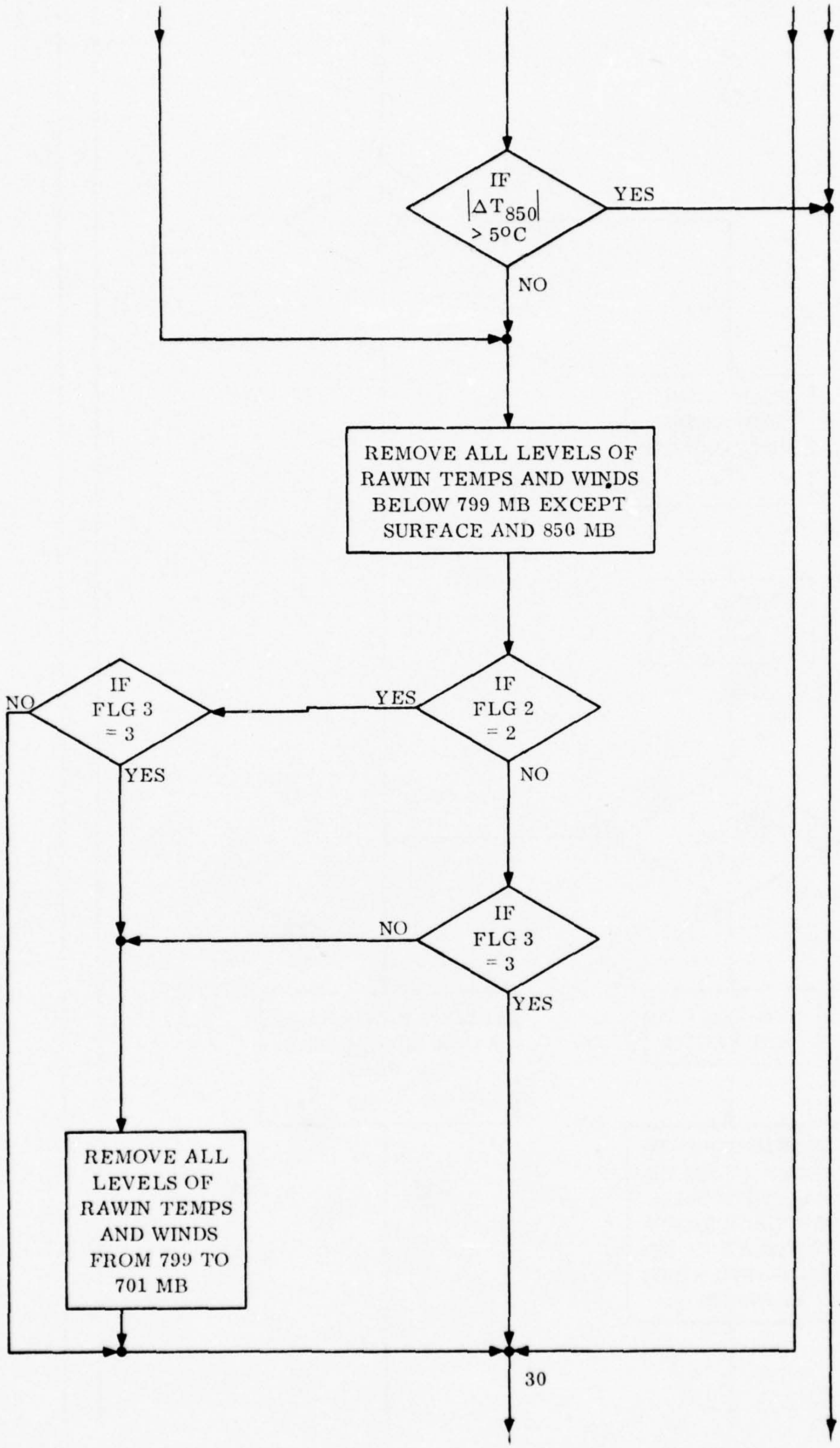

IS THERE A STATION
DETERMINE MOST RECENT GWC BASE TIME AT THIS STATION
IS THERE GWC OBS FOR BASE TIME
IS THERE GWC DATA FOR 1000, 850 AND 700 MB FOR RELEASE TIME
IS THERE RAWIN DATA AT SURFACE, 850 AND 700 MB

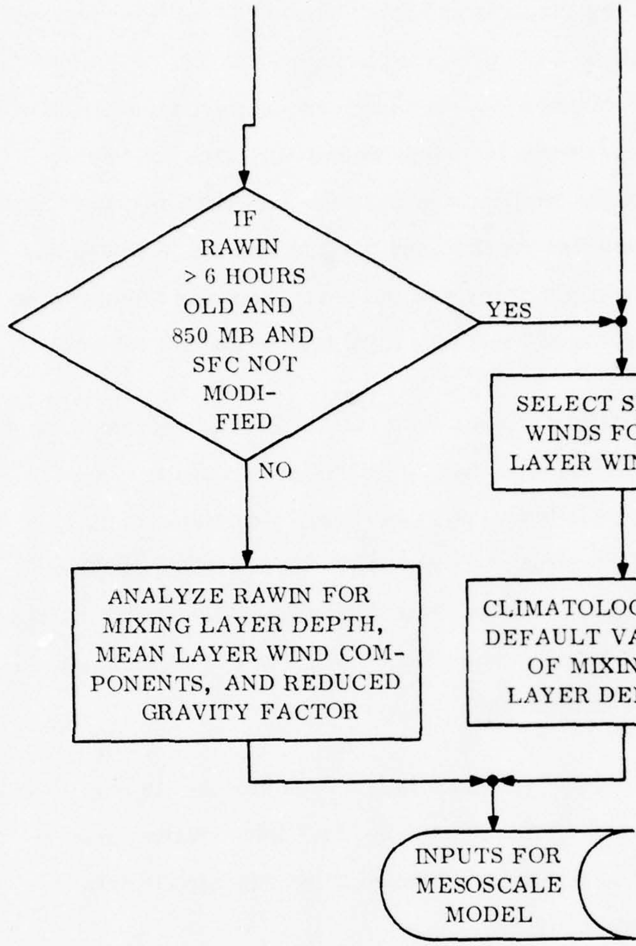

SELECT GWC DATA AT 700 MB (T'700) AND RAWIN DATA AT 700 MB
FORM ΔT700 = T'700 - T700
IF |ΔT700| > 5°C
YES
NO
IF ΔT700 > 0 SET FLG2=2
REPLACE T700 BY T'700 AND RAWIN WINDS BY GWC WINDS
SET FLG4
READ SURFACE OBS INVENTORY STARTING WITH MOST CURRENT DATA BLOCK
YES
IF FLG 1
NO


DOES BLOCK HAVE DATA ≤ 3 HOURS OLD
NO
YES
DEFINE LIST OF SURFACE STATIONS < 50 KM FROM RELEASE POINT
ORDER STATIONS BY INCREASING DISTANCE FROM SOURCE
INCREMENT STATION INDEX
IS LIST EXHAUSTED
YES
NO
DECREMENT TIME INDEX STARTING WITH MOST CURRENT TIME FOR STATION INDEX


IS DATA ≤ 3 HOURS OLD
NO
YES
IS THERE PRESS, TEMP DATA
NO
YES
DEFINE $\Delta P = P_S - P$ WHERE P_S = SFC STATION PRESSURE, P = RAWIN SFC PRESSURE
IF $\Delta P > 10$ MB
YES
NO
REPLACE RAWIN SFC TEMP WITH STAT TEMP T_S. REPLACE WINDS AT SURFACE


IF FLG 4
YES
NO
SELECT GWC TEMP AT 850 MB (T'_{850}) AND T_s
FORM $\Delta T'_{850} = T'_{850} - T_s$
IF $|\Delta T'_{850}| > 5^oC$
YES
NO
IF $\Delta' T_{850} > 0$, SET FLG3=3
SELECT RAWIN TEMP AT 850 MB T_{850} AND T_s. FORM $\Delta T_{850} = T_{850} - T_s$
REPLACE RAWIN TEMP AT 850 MB T_{850} WITH T'_{850}. REPLACE RAWIN WINDS AT 850 MB WITH GWC WINDS AT 850 MB


IF $|\Delta T_{850}| > 5^{o}C$
YES
NO
REMOVE ALL LEVELS OF RAWIN TEMPS AND WINDS BELOW 799 MB EXCEPT SURFACE AND 850 MB
IF FLG 3 = 3
NO
YES
IF FLG 2 = 2
YES
NO
IF FLG 3 = 3
NO
YES
REMOVE ALL LEVELS OF RAWIN TEMPS AND WINDS FROM 799 TO 701 MB
30


IF RAWIN > 6 HOURS OLD AND 850 MB AND SFC NOT MODIFIED
YES
NO
SELECT SFC WINDS FOR LAYER WINDS
ANALYZE RAWIN FOR MIXING LAYER DEPTH, MEAN LAYER WIND COMPONENTS, AND REDUCED GRAVITY FACTOR
CLIMATOLOGICAL DEFAULT VALUES OF MIXING LAYER DEPTH
INPUTS FOR MESOSCALE MODEL

temperatures and winds at the 700 millibar are replaced if the GWC predicted temperature for the hour of interest is within 5 degrees Celsuis; rawinsonde temperatures and winds at 850 millibars are replaced if the temperature from the reporting surface station nearest the time and point of release is within 5 degrees Celsuis of the GWC predicted temperature at the 850 millibar level for the hour of interest

- Surface temperature and wind data from the closest station less than 50 kilometers from the release point, less than three-hours old, and with a surface pressure within 10 millibars of the surface pressure from the rawinsonde observation, is used to modify the surface rawinsonde observation when it is more than three-hours old

- Surface wind speeds from the closest station to the point of release are used as default values for the mean-layer wind speed if the rawinsonde data is not analyzed

Climatological estimates of the mixing layer depth shown in Table 3-1 are used as default values where the rawinsonde data is not available or is not analyzed for some reason. These estimates of the median mixing depths as a function of month and time-of-day are based on the work of Norton and Hoidale (1975), who analyzed 8,236 rawinsonde observations made at WSMR between 1961 and 1972 using the technique described by Holzworth (1967).

As shown in Figure 3-1, the rawinsonde is analyzed to determine the mixing-layer depth, mean-layer wind direction and speed and reduced gravity

TABLE 3-1

CLIMATOLOGICAL ESTIMATES OF THE MEDIAN SURACE MIXING DEPTH H_m IN METERS (AFTER NORTON AND HOIDALE, 1975)

Local Time (Hour Ending)	Month											
	Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
0700	30	30	30	50	30	30	30	30	30	30	30	30
0800	30	30	30	107	110	139	103	35	30	30	30	30
0900	30	30	114	326	467	403	329	238	129	89	30	30
1000	64	122	387	740	865	691	688	615	490	271	165	96
1100	232	329	832	1255	1298	1274	1034	1042	933	539	400	239
1200	434	639	1247	1761	1907	2053	1342	1431	1343	873	593	369
1300	645	1054	1518	2177	2521	2654	1868	1840	1708	1217	792	533
1400	822	1363	1804	2513	2915	2983	2388	2347	1896	1554	1025	734
1500	907	1543	2145	2872	3195	3162	2503	2749	1940	1711	1126	718
1600	753	1586	2276	2843	3292	3144	2116	2484	2038	1344	1039	435
1700	386	1148	1835	1954	2727	3065	1279	1572	1762	696	669	206
1800	102	436	946	824	1578	2606	516	816	932	280	231	102
1900	30	68	255	229	549	1466	191	398	252	181	71	35
2000	30	111	30	63	77	427	172	115	30	164	45	30
2100	30	142	30	30	30	30	153	30	30	77	30	30
2200-0600	30	30	30	30	30	30	30	30	30	30	30	30

factor. In developing an automated procedure for determining the mixing-layer depth, we studied the use of four methods for establishing that a surface-based or elevated inversion was indicated by the rawinsonde observations. The simple criteria associated with the four methods for determining the base of an inversion were:

Method	Criteria
1	$\frac{\Delta \theta}{\Delta z} > .01$
2	$\frac{\Delta T}{\Delta z} \geq 0$
3	$\frac{\Delta \theta_w}{\Delta z} \geq 0$
4	$\frac{\Delta T_v}{\Delta z} \geq -5 \times 10^{-4}$

where

$\frac{\Delta \theta}{\Delta z}$ = lapse rate of potential temperature

$\frac{\Delta T}{\Delta z}$ = lapse rate of temperature

$\frac{\Delta \theta_w}{\Delta z}$ = lapse rate of wet-bulb potential temperature

$\frac{\Delta T_v}{\Delta z}$ = lapse rate of virtual temperature

The computer program incorporating these criteria was used to analyze twice daily rawinsonde observations from Salt Lake City, Utah for the first five days of each month during the period June 1972 through May 1973 and for all the month of January 1973. The heights of the bases of stable layers obtained from the computer program were then qualitatively compared with those selected by meteorologists from plots of the data. All the criteria produced bases of stable layers in agreement with those selected by the meteorologists for cases in which clearly-defined stable layers were present. In our view, however, the criteria based on the lapse rate of virtual temperature (Method 4) performed more consistently than the other criteria when stable layers were not as clearly defined and for all seasons of the year. For this reason, the procedure finally used in the program is based on this criteria.

The program accepts the raw or modified rawinsonde data and begins the selection process by checking

$$\frac{\Delta T_{v,n}}{\Delta z_n} = \frac{T_{v,n+1} - T_{v,n}}{z_{n+1} - z_n} \tag{3-1}$$

in the first height interval above the surface. If the criterion for the base of a stable layer is met $\left(\text{e.g., if } \Delta T_{v,1}/\Delta z_1 \geq -5 \times 10^{-4}\right)$ and the height interval Δz is greater than or equal to 100 meters, the program defines a surface-based inversion and sets, by default, the surface mixing depth to 30 meters. If Δz_1 is less than 100 meters but $\Delta T_{v,1}/\Delta z_1 \geq -5 \times 10^{-4}$, the program then checks $\Delta T_{v,2}/\Delta z_2$. If $\Delta T_{v,2}/\Delta z_2 \geq -5 \times 10^{-4}$ and $\left(\Delta z_1 + \Delta z_2\right) \geq 100$ meters, the program checks the quantity

$$\left(\frac{\Delta T}{\Delta z}\right)' = \frac{T_{v,n+2} - T_{v,n}}{z_{n+2} - z_n} = \frac{T_{v,3} - T_{v,1}}{z_3 - z_1} \qquad (3\text{-}2)$$

If $(\Delta T/\Delta z) \geq -5 \times 10^{-4}$, the program again defines a surface-based inversion. If no surface based inversion is found, the program increments n and proceeds through the same operations until the base of an elevated inversion is found or until z_{n+1} exceeds 3000 meters AGL. If z_{n+1} exceeds 3000 meters AGL, the surface mixing depth is defaulted to 3000 meters.

After the depth of the surface mixing-layer H_m is established, the program analyzes the rawinsonde data between the surface and H_m to obtain estimates of the mean-layer wind components and the reduced gravity factor using the expressions:

$$u_{H_m} = \frac{\sum_{n=1}^{H_m} u_n \left(z_{n+1} - z_n\right)}{\sum_{n=1}^{H_m} \left(z_{n+1} - z_n\right)} \qquad (3\text{-}3)$$

$$v_{H_m} = \frac{\sum_{n=1}^{H_m} v_n \left(z_{n+1} - z_n\right)}{\sum_{n=1}^{H_m} \left(z_{n+1} - z_n\right)} \qquad (3\text{-}4)$$

$$g' = g\left[1 - \frac{\left(\sum_{n=1}^{H_m} T_n \left(z_{n+1} - z_n\right) \Big/ \sum_{n=1}^{H_m} \left(z_{n+1} - z_n\right)\right)}{T_{I,\max}}\right] \qquad (3\text{-}5)$$

where $T_{I,max}$ is the maximum temperature in the stable layer above H_m. In the program, g' is restricted so that if it exceeds 0.3 it is set to 0.3 and if less than 0.1 is set to 0.1.

3.2 DIFFUSION MODEL ROUTINE

The meteorological and source parameters required by the diffusion models are given in Table 3-2. Of the meteorological parameters shown in the table, the user need only supply the program with values of r and A if they differ from the default values of 1 and 0, respectively. The program user must supply all the source input parameters to the program and, in addition, denote whether the source is instantaneous or continuous. For program test purposes, the program now contains two default continuous sources corresponding to the American Smelting and Refining Company stacks associated with their lead and copper refineries in El Paso, Texas. The source inputs for these stacks were supplied by WSMR. Default values of V_s and k equal to zero are used in the program.

Since the turbulence parameters $\sigma'_{A,n}$ and $\sigma'_{E,n}$, their power-law coefficients for height dependency m and n, and the power-law coefficient p_n are not routinely measured at meteorological stations, a means of assigning appropriate values to these meteorological parameters was required. Swanson and Cramer (1965) made a comprehensive study of the height dependence of σ_A as a function of wind speed and time-of-day using measurements from a 62-meter tower at WSMR. The decision was made to use the results of this study in conjunction with a stability classification system, similar to the Pasquill definition of stability categories (Turner, 1964) used by the Environmental Protection Agency, to select and assign values to these parameters. Figure 3-2 is a schematic diagram showing the logic sequence used by the program in defining the parameters and their use in the diffusion models.

TABLE 3-2

METEOROLOGICAL AND SOURCE INPUTS
FOR THE DIFFUSION MODELS

Parameter	Definition
(a) Meteorological Inputs	
NR	Net radiation index
$H_{m,n}$	Depth of the surface mixing layer at the n^{th} point in the trajectory
$\bar{\bar{u}}_n$	Mean layer wind speed in the surface mixing layer at the n^{th} point on the trajectory
$\sigma'_{A,n}$	Standard deviation of the wind azimuth angle at the n^{th} point on the trajectory
m	Power-law coefficient used to describe the height dependency of $\sigma_{A,n}$ in the surface mixing layer
$\sigma'_{E,n}$	Standard deviation of the wind elevation angle at the n^{th} point on the trajectory
n	Power-law coefficient used to describe the height depending of $\sigma_{E,n}$ in the surface mixing layer
p_n	Power-law coefficient used to describe the height dependency of wind speed in the surface mixing layer
r	Fraction of material reflected at the surface
Λ	Washout coefficient or fraction of material removed by precipitation scavenging per unit time

TABLE 3-2 (Continued)

Parameter	Definition
(b) Source Inputs	
x, y, h	Coordinates of the source
Q	Source Strength
T_c	Stack exit temperature (^oK) for the continuous source
r_s	Inner radius of the stack at the exit
w	Stack exit velocity
V_s	Settling velocity for material in a given size category
k	Decay coefficient or fraction of material lost per unit time

Key features of the logic sequence shown in Figure 3-2 are that:

- Solar altitudes for the release point and time of request are calculated using an algorithm suggested by Woolf (1968)

- The program then seeks the most recent cloud and ceiling data from the nearest reporting station to the release point

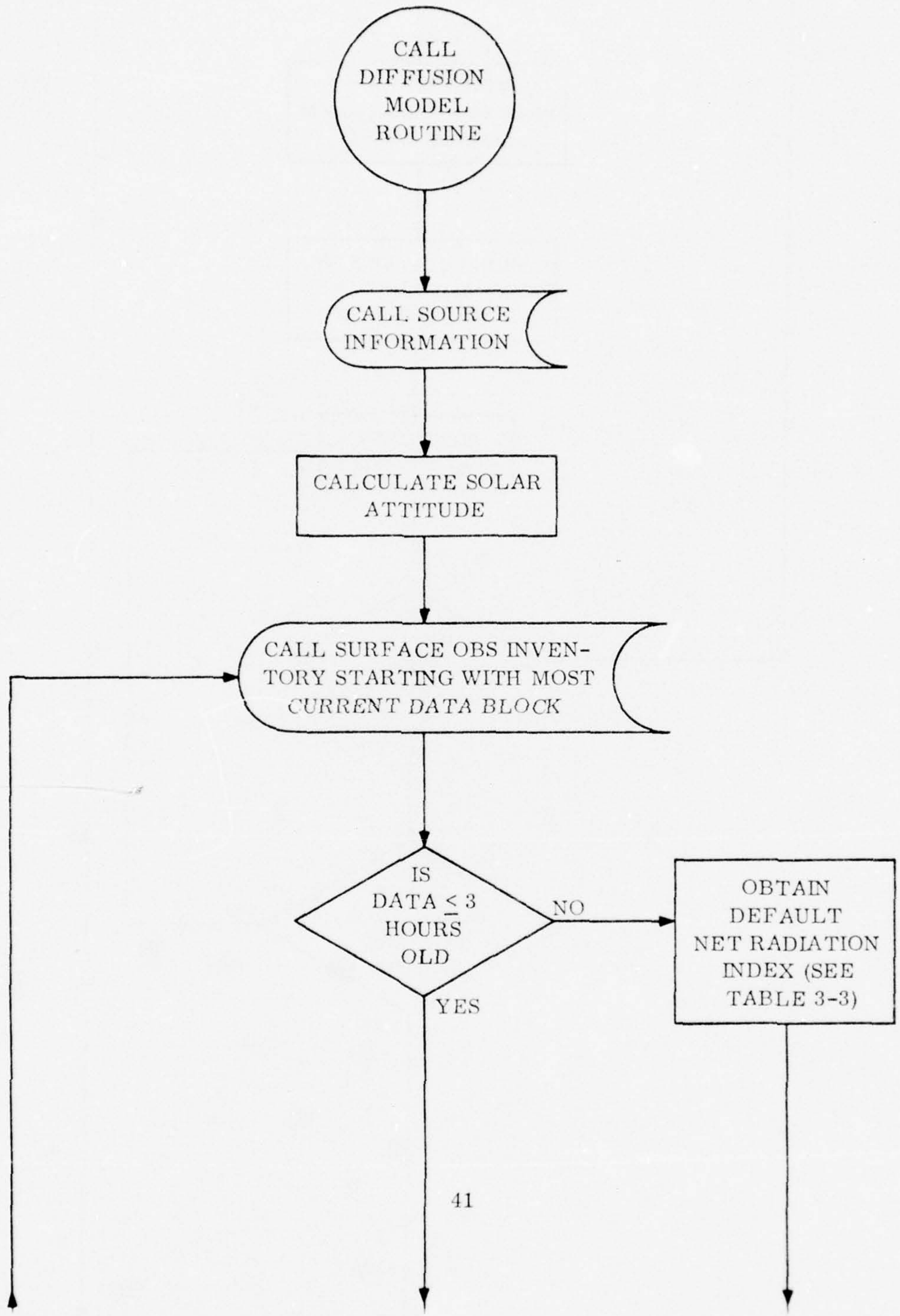
- If there are no stations reporting cloud and ceiling data within a 50-kilometer radius from the release point or the data is more than three-hours old, the program selects a default net radiation index from Table 3-3

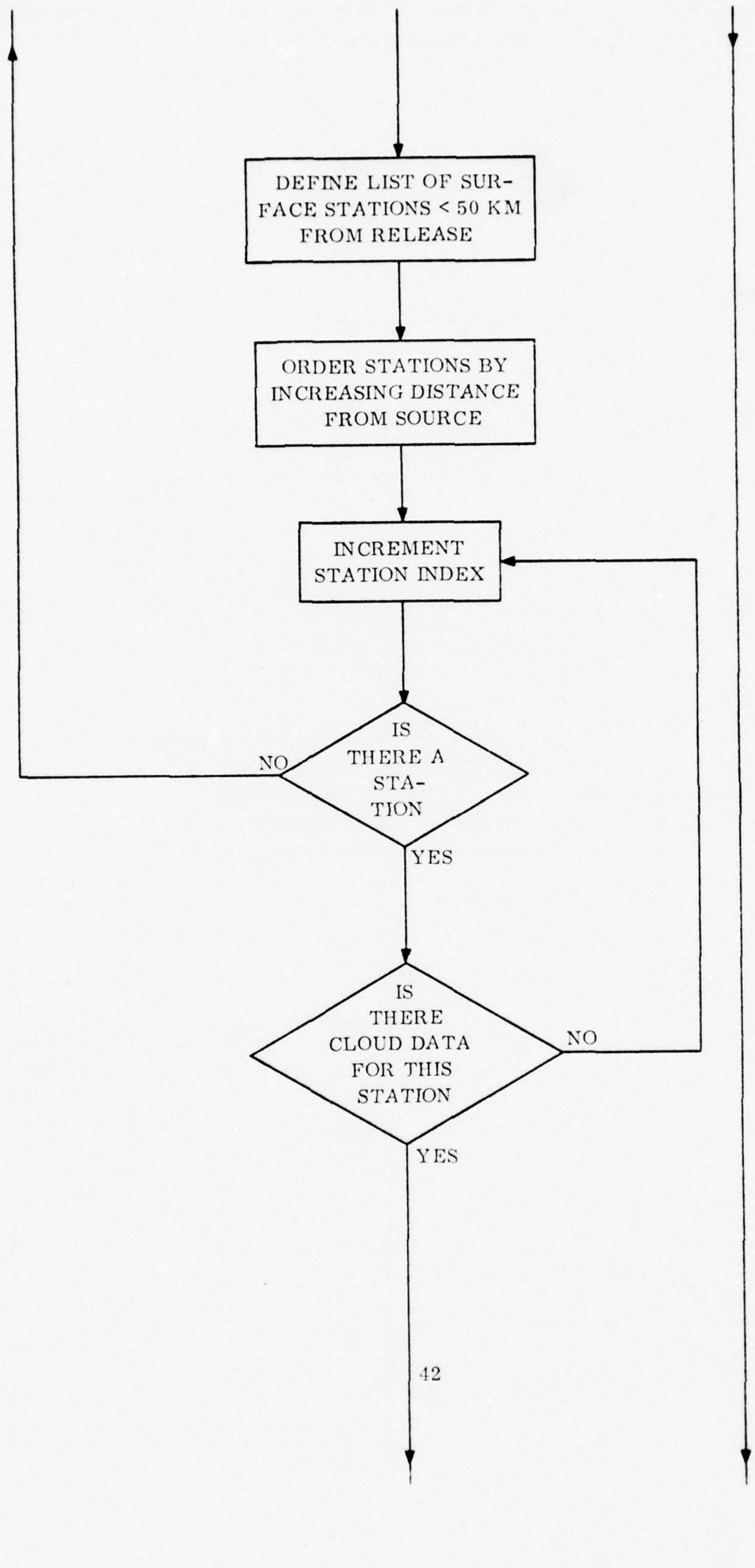
- When cloud data is available, the program calculates an initial net radiation index based on solar altitude (also see Table 3-3) and adjusts the index based on the time-of-day total cloud cover, and ceiling height

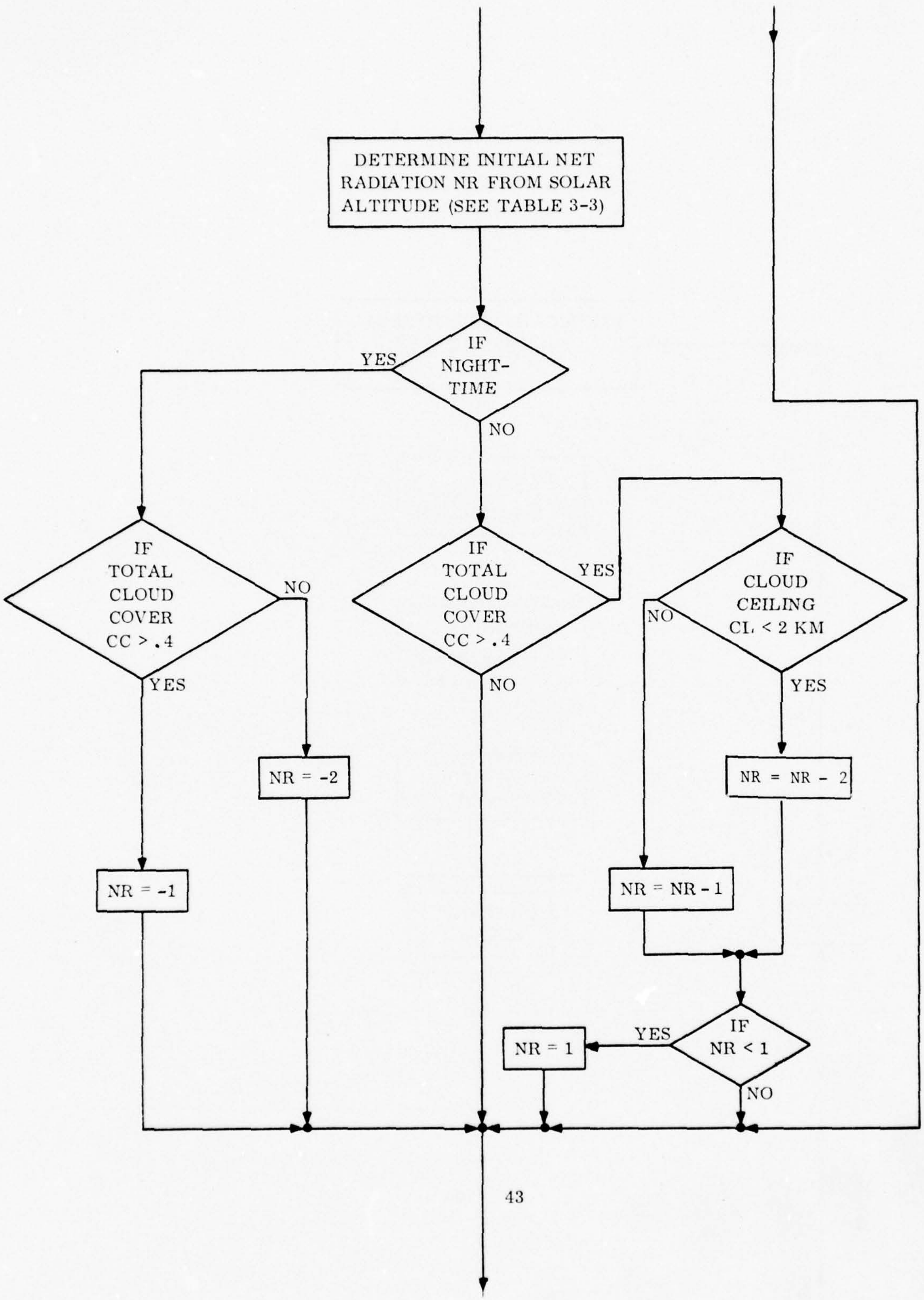
- The net radiation index and the wind speeds from the mesoscale wind-field prediction model for the release point and points along the cloud trajectory are used to select turbulent input parameters and wind power-law coefficients for the diffusion models

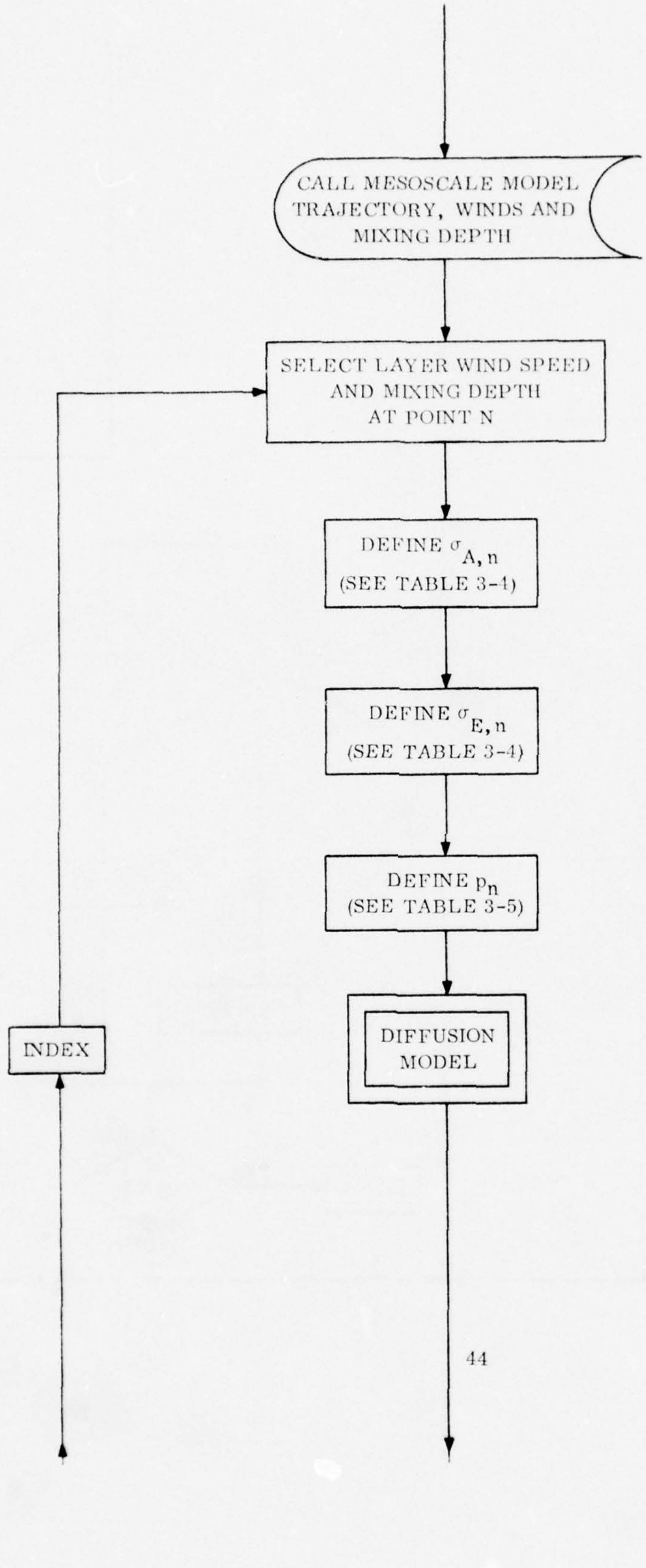
- Values of the surface mixing depth $H_{m,n}$ for each n^{th} point along the cloud trajectory are obtained from the mesoscale wind-field solution using the interpolation procedure given by Equation (2-5)

Figure 3-2. Schematic diagram illustrating procedures for obtaining inputs to the diffusion models.




DEFINE LIST OF SURFACE STATIONS < 50 KM FROM RELEASE
ORDER STATIONS BY INCREASING DISTANCE FROM SOURCE
INCREMENT STATION INDEX
IS THERE A STATION
NO
YES
IS THERE CLOUD DATA FOR THIS STATION
NO
YES


DETERMINE INITIAL NET RADIATION NR FROM SOLAR ALTITUDE (SEE TABLE 3-3)
IF NIGHT-TIME
YES
NO
IF TOTAL CLOUD COVER CC > .4
NO
YES
IF TOTAL CLOUD COVER CC > .4
YES
NO
IF CLOUD CEILING CL < 2 KM
NO
YES
NR = -2
NR = NR - 2
NR = -1
NR = NR - 1
IF NR < 1
YES
NO
NR = 1



CALL MESOSCALE MODEL TRAJECTORY, WINDS AND MIXING DEPTH
SELECT LAYER WIND SPEED AND MIXING DEPTH AT POINT N
DEFINE $\sigma_{A,n}$ (SEE TABLE 3-4)
DEFINE $\sigma_{E,n}$ (SEE TABLE 3-4)
DEFINE p_n (SEE TABLE 3-5)
DIFFUSION MODEL
INDEX

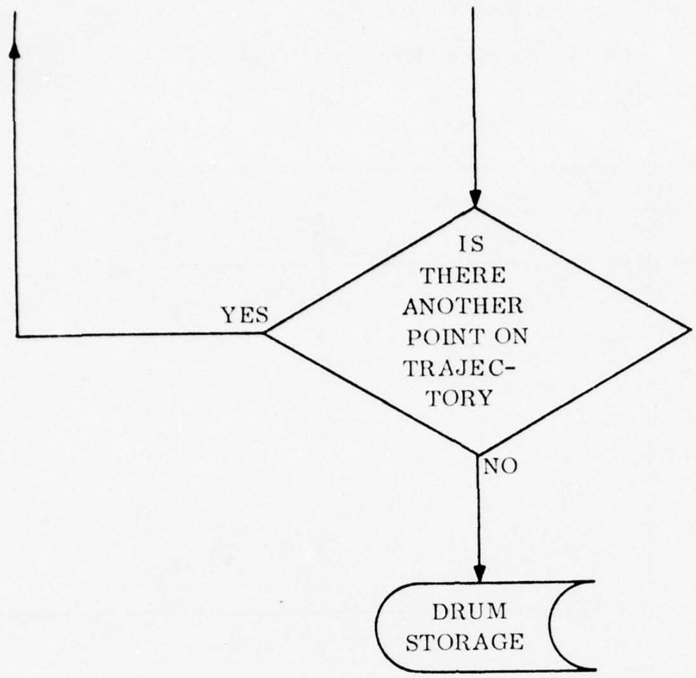

IS THERE ANOTHER POINT ON TRAJEC- TORY
YES
NO
DRUM STORAGE

TABLE 3-3

NET RADIATION INDICES

Solar Altitude η In Degrees	Daytime Index	Default Index
$\eta > 60$	4	3
$35 < \eta \leq 60$	3	2
$15 < \eta \leq 35$	2	1
$0 < \eta \leq 15$	1	0
Nighttime	N/A	-1

Table 3-4(a) shows values of the ten-minute standard deviations of the wind azimuth angle in degrees $\left(\sigma_A \{\tau = 10 \text{ min}\}\right)$ and associated values of the power-law coefficient m for describing σ_A variation with height above the surface based on the study of Swanson and Cramer. Values of the standard deviation of the wind elevation angle in degrees and the power-law coefficient n describing the variation of σ_E with height above the surface are shown in Figure 3-4(b). These values of σ_E were assigned under the assumption that (after Cramer, et al., 1972) turbulence is isotropic for short averaging times and that the one-fifth power law can be used as a scale factor to adjust σ_A for averaging times less than 10 minutes. Thus,

$$\sigma_E \simeq \sigma_A \{\tau = 10 \text{ min}\} \left(\frac{2.5}{600}\right)^{1/5} \qquad (3\text{-}6)$$

$$\simeq \sigma_A \{\tau = 10 \text{ min}\}/2.99 \qquad (3\text{-}7)$$

The values of the power-law coefficient n were assigned under the assumptions that:

- Under stable conditions (NR = -1, -2)

$$\sigma_{E,z} \sim \sigma_{E,z_R} \left(\frac{z}{z_R}\right)^{-p_n} \qquad (3\text{-}8)$$

- Under near-neutral conditions (NR = 0, 1)

TABLE 3-4

TURBULENCE PARAMETERS FOR TRANSPORT/DIFFUSION MODELS

(a) Ten-Minute Standard Deviations of the Wind Azimuth Angle in Degrees (σ_A) and Corresponding Values of the Power-Law Exponent m

Wind Speed at 5 Meters (m/sec)	Net Radiation Index													
	4		3		2		1		0		-1		-2	
	σ_A	m	σ_A	m	σ_A	m	σ_A	m	σ_A	m	σ_A	m	σ_A	m
$\bar{u} < 1$	26	-.109	26	-.109	21	-.133	15	-.112	11	-.120	8.6	-.189	8.6	-.189
$1 \leq \bar{u} < 3$	26	-.109	22	-.128	16	-.123	11	-.112	10	-.120	8.6	-.189	8.6	-.189
$3 \leq \bar{u} < 5$	19	-.122	16	-.123	13	-.103	9	-.120	9	-.134	7.8	-.207	7.8	-.189
$5 \leq \bar{u} < 7$	14	-.105	11	-.117	11	-.117	8	-.148	7	-.126	6.7	-.287	6.7	-.189
$7 \leq \bar{u}$	9	-.100	9	-.117	8	-.110	8	-.148	7	-.126	6.7	-.287	6.7	-.189

(b) Standard Deviation of the Wind Elevation Angle In Degrees (σ_E) and Corresponding Values of the Power-Law Exponent n

Wind Speed at 5 Meters (m/sec)	Net Radiation Index													
	4		3		2		1		0		-1		-2	
	σ_E	n	σ_E	n	σ_E	n	σ_E	n	σ_E	n	σ_E	n	σ_E	n
$\bar{u} < 1$	8.7	.10	8.7	.10	7.0	.07	5.0	0	3.7	0	2.9	-.20	2.9	-.30
$1 \leq \bar{u} < 3$	8.7	.10	7.4	.13	5.3	.07	3.7	0	3.3	0	2.9	-.20	2.9	-.25
$3 \leq \bar{u} < 5$	6.3	.15	5.3	.15	4.3	.13	3.0	0	3.0	0	2.6	-.20	2.6	-.20
$5 \leq \bar{u} < 7$	4.7	.20	3.7	.20	3.7	.15	2.7	0	2.7	0	2.2	-.15	2.2	-.15
$7 \leq \bar{u}$	3.0	.25	3.0	.20	2.7	.20	2.7	0	2.7	0	2.2	-.15	2.2	-.15

$$\sigma_{E,z} \sim \sigma_{E,z_R} \tag{3-9}$$

- Under unstable conditions (NR = 2, 3, 4)

$$\sigma_{E,z} \sim \sigma_{E,zR} \left(\frac{z}{z_R}\right)^{0.3-p_n} \tag{3-10}$$

The values of the wind power-law coefficient p_n as a function of net radiation index and mean layer wind speed used in the program are shown in Table 3-5. These values are similar to those suggested for use at WSMR by Swanson and Hoidale (1962) and for Dugway Proving Ground by Cramer, et al., (1972).

The plume-rise models described in Section 2.2.2 above require estimates of the ambient air temperature and, under stable conditions, the lapse-rate of potential temperature. If surface meteorological data are available, the program automatically selects the ambient air temperatures according to the logic shown in Figure 3-2 and the lapse-rate of potential temperature is calculated from the rawinsonde sounding established using the logic described in Figure 3-1. If no surface data is available, the default values of surface temperature shown in Table 3-6 are used. The temperatures in the table are based on the monthly normal and maximum temperatures at El Paso extracted from the report "Climatic Atlas of the United States" published by the U. S. Department of Commerce in June 1968. The values shown for a net radiation index of -1 and -2 correspond to the normal minimum temperature for each month. Similarly, the values shown for net radiation indices of 0 and 1 and for indices of 2 and 3 respectively correspond to the normal average and normal maximum temperatures for each month. Finally, the surface temperatures for a net radiation index of -4 were obtained by taking an average of

TABLE 3-5

WIND POWER-LAW EXPONENT p FOR TRANSPORT/DIFFUSION MODELS

Mean Layer Wind Speed $\bar{u}$ (m/sec)	Net Radiation Index						
	4	3	2	1	0	-1	-2
$\bar{u} < 1$	.2	.2	.20	.20	.20	.20	.3
$1 \le \bar{u} < 3$	.2	.17	.20	.20	.20	.20	.25
$3 \le \bar{u} < 5$	.15	.15	.17	.17	.20	.20	.20
$5 \le \bar{u} < 7$	.10	.10	.15	.15	.15	.15	.15
$7 \le \bar{u}$	.05	.10	.10	.10	.15	.15	.15

the recorded maximum temperature and the normal maximum temperature at El Paso for each month.

If no rawinsonde data are available, the program selects default values of the lapse rate of potential temperature from Table 3-7 based on the net radiation index and the calculated wind speed at stack height. The potential temperature lapse rates shown in Table 3-6 are based on our experience in applying the Briggs stable plume rise formula in predicting plume rise from industrial stacks located in relatively warm and dry climates. No values of the default potential temperature lapse rates are given for wind speeds greater than 5 meters per second and for net radiation indices greater than 1 because plume rise is calculated from Equation (2-29) rather than Equation (2-32) under these conditions.

TABLE 3-6

DEFAULT VALUES OF SURFACE TEMPERATURE T_s (^oK)

Month	Net Radiation Index			
	4	3, 2	1, 0	-1, -2
Jan	292	286	279	272
Feb	296	290	282	275
Mar	300	294	286	277
Apr	303	299	290	282
May	308	304	295	287
Jun	312	308	300	292
Jul	312	308	301	294
Aug	310	307	300	293
Sep	308	304	297	289
Oct	303	299	291	283
Nov	297	292	284	275
Dec	293	287	280	272

TABLE 3-7

DEFAULT VALUES OF THE LAPSE RATE OF POTENTIAL TEMPERATURE $\bar{\Phi}$ (deg K/m)

Wind Speed $\bar{u}\{h\}$ (m sec^{-1})	Net Radiation Index			
	1	0	-1	-2
0 - 1.4	0.005	0.015	0.030	0.040
1.5 - 2.9	0.003	0.010	0.020	0.030
3.0 - 4.9	0.001	0.005	0.015	0.020

SECTION 4

ANALYSIS, APPLICATION AND GRAPHICS PHASES OF EPAMS

Figure 4-1 shows the executive components, in the form of a block diagram, of the EPAMS System. The EPAMS executive system manager, labeled XECAMS in Figure 4-1, is where all processing begins and ends. The executive systems manager on option passes control to the supervisors labeled ANALIZ, APPLY and PLTPRG which were designed and developed by the H. E. Cramer Company, Inc. to accomplish the technical objectives of the contract.

The analysis supervisor ANALIZ controls all processing in the Analysis Phase of EPAMS and passes control to the mixing layer and wind-field analysis foreman MIXLYR or to other meteorological analysis routines. The operational elements of MIXLYR are described in Section 4.1 below. The application supervisor APPLY passes control on option to the transport/diffusion application foreman TRYDIF or to other application routines. The operational elements of TRYDIF are described in Section 4.2. At the completion of either the Analysis or Application Phases, or at the completion of both phases, program control is returned to XECAMS. The executive systems manager can then address, on option, the graphics supervisor PLTPRG to obtain either printed or plotted solutions, or both types of solutions, developed in the application routines. Further details of the graphics supervisor PLTPRG are described in Section 4.3 below.

4.1 ANALYSIS PHASE OF EPAMS

As shown in Figure 4-1, the foreman MIXLYR of the Analysis Phase of EPAMS is comprised of the two operational elements MXOBAN and SOLVMX. The operational subroutines of MXOBAN are shown in the form of a block diagram in Figure 4-2. The operational element MXOBAN controls the analysis of all meteor-

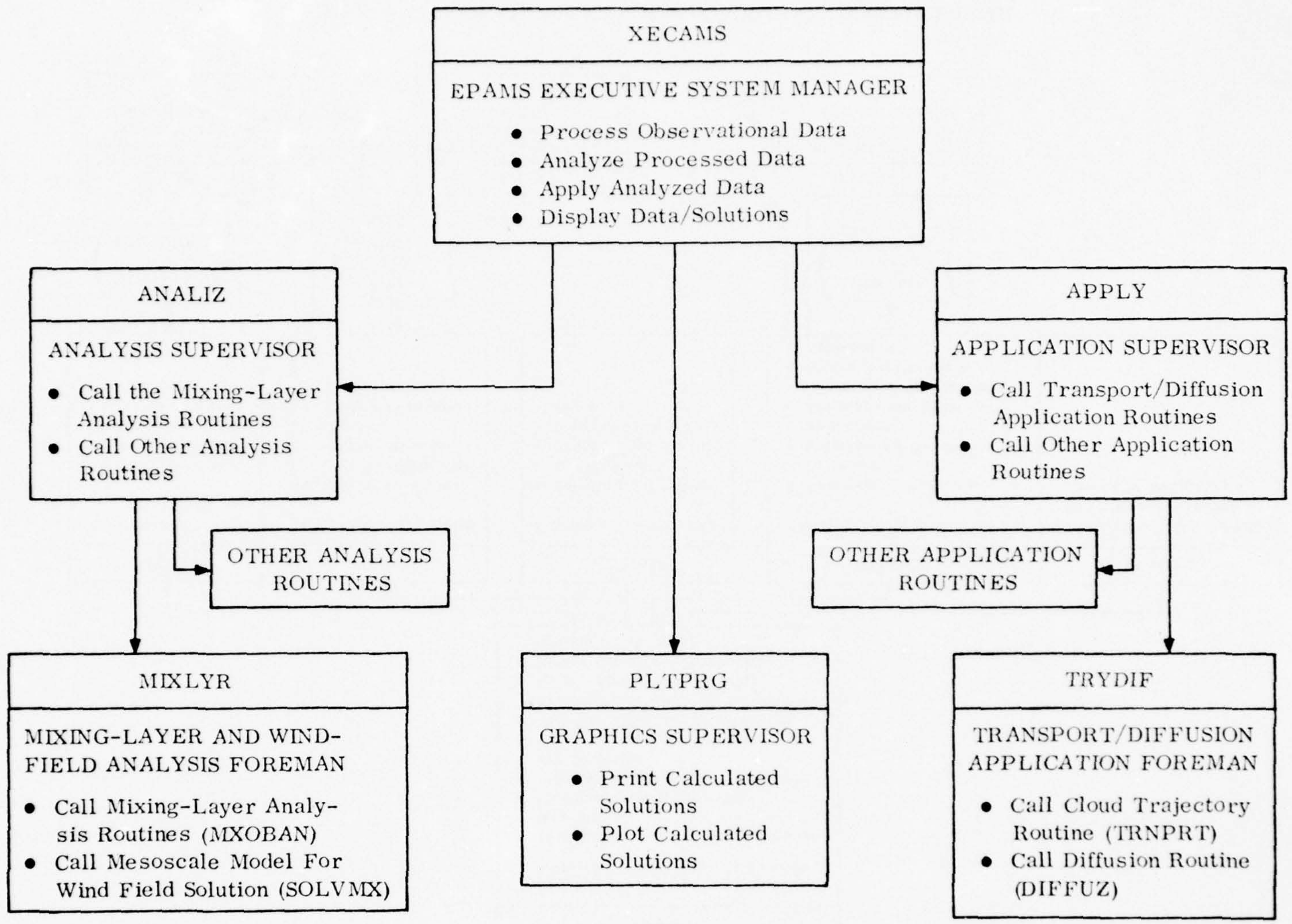


Figure 4-1. Executive System and Supervisors of the Analysis, Application and Graphics Phases of EPAMS.

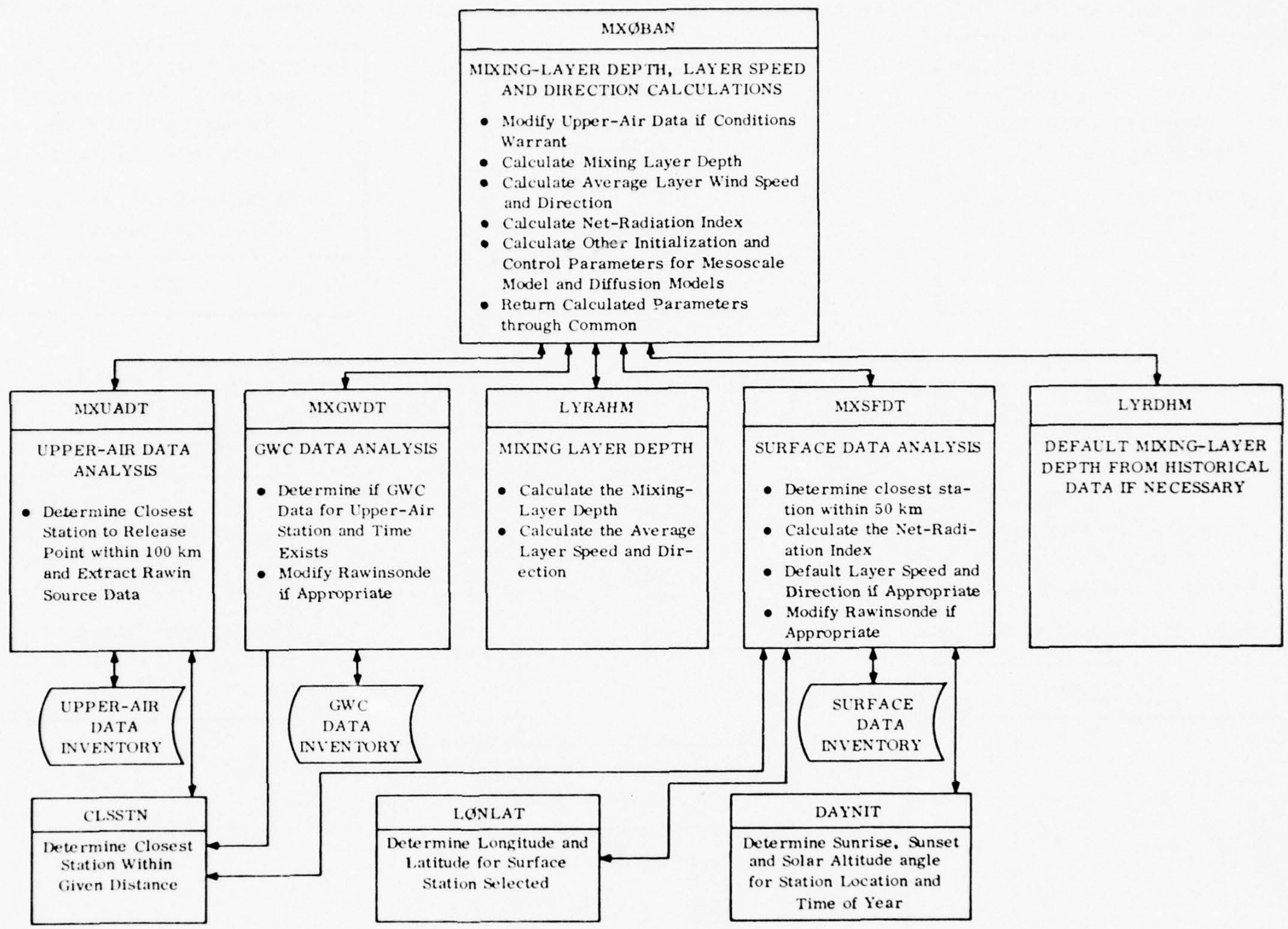


Figure 4-2. Operational Elements of MXØBAN Analysis Phase.

ological data and passes the results of the data analysis to the subroutine SOLVMX at the completion of execution. The EPAMS System user requests transport and diffusion information for a given time and source, which is then passed to MXOBAN via XECAMS. The MXOBAN mixing-layer analysis routine then proceeds to analyze rawinsonde, GWC and surface meteorological data according to the logic process described in Section 3.1. The subroutine MXUADT obtains rawinsonde data from the EPAMS data base for the desired solution time and source location. GWC prediction data and surface weather observations for use in the analysis are obtained from the data base via the subroutines MXGWDT and MXSFDT. These subroutines also use the GWC and surface data to modify the rawinsonde data if appropriate. The subroutine MXSFDT also calculates the net-radiation index for the surface meteorological station using information obtained from the LONLAT and DAYNIT subroutines. The rawinsonde data is then used in the subroutine LYRAHM to calculate the depth of the surface mixing layer and the mean wind components in the layer as described in Section 3.1. If insufficient data are available for calculating the mixing-layer depth or wind components, default data is supplied via the LYRDHM and MXSFDT subroutines. Control is then returned to the operational element MIXLYR and processing continues by branching to the subroutine SOLVMX.

A block diagram showing operational subelements of SOLVMX is given in Figure 4-3. Processing begins through examination of the mesoscale model input parameters passed to SOLVMX from MIXLYR to determine which of two procedures are to be used in initializing the mesoscale model. Thus, the input parameters from MIXLYR are compared in the subroutine MXSTRT with the input parameters used to generate the last solution of the mesoscale model resident in the EPAMS data base. The mesoscale model is initialized with the new inputs from MXOBAN if the mean layer wind speeds and elevation of the mixing depth above mean sea level (MSL) differ by more than twenty-five percent and the mean wind directions differ by more than ten degrees (cold start). The initial layer depth and momentum components for a cold start are given by the expressions

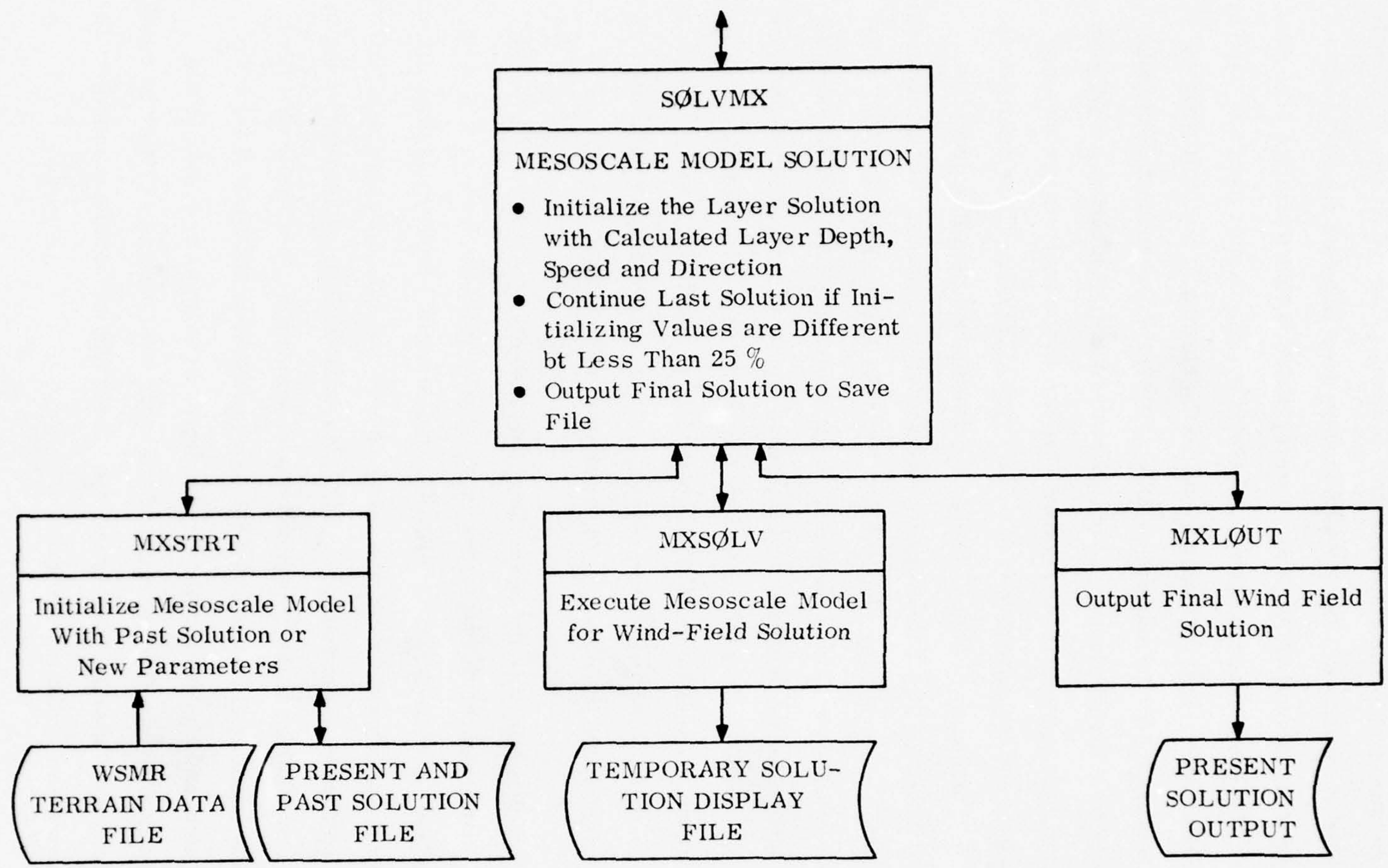


Figure 4-3. Operational Elements of SØLVMX Analysis Phase.

$$H_{m;i,j} = H'_n - h'_{i,j} \tag{4-1}$$

$$M_{i,j} = U_n\left(H'_n - h'_{min}\right) \tag{4-2}$$

$$N_{i,j} = V_n\left(H'_n - h'_{min}\right) \tag{4-3}$$

where

H'_n = mixing depth (MSL) calculated for the present hour

$h'_{i,j}$ = terrain elevation (MSL) at each solution matrix grid point

U_n = easterly component of the mean layer wind for the present hour

h'_{min} = minimum terrain elevation (MSL) in the solution matrix

V_n = northerly component of the mean layer wind for the present hour

If the differences are less than those specified above, subroutine MXSTRT calculates initialization parameters for a "hot" start from the expressions

$$H_{m;i,j} = \left(H'_{i,j} - h'_{i,j}\right) + H'_n - H'_o \tag{4-4}$$

$$M_{i,j} = \left(H'_{i,j} - h'_{i,j}\right) U_{i,j} + \left(H'_o - h'_{min}\right)\left(U_n - U_o\right) + U_n\left(H'_n - H'_o\right) \qquad (4\text{-}5)$$

$$N_{i,j} = \left(H'_{i,j} - h'_{i,j}\right) V_{i,j} + \left(H'_o - h'_{min}\right)\left(V_n - V_o\right) + V_n\left(H'_n - H'_o\right) \qquad (4\text{-}6)$$

where

$H'_{i,j}$ = mixing depth (MSL) at each solution matrix grid point for the previous wind-field solution

H'_o = mixing depth (MSL) used to initialize the previous wind-field solution

$U_{i,j}$ = easterly component of the mean layer wind at each solution matrix grid point for the previous wind-field solution

U_o = easterly component of the mean layer wind used to initialize the previous wind-field solution

$V_{i,j}$ = northerly component of the mean layer wind at each solution matrix grid point for the previous wind-field solution

V_o = northerly component of the mean layer wind used to initialize the previous wind-field solution

After the initialization procedure has been completed, the input parameters are passed to the subroutine MXSOLV via SOLVMX for execution of the mesoscale wind-field model described in Section 2.1.

Subroutine MXSOLV will allow the mesoscale model to execute for a maximum model time of approximately 7200 seconds for a cold start and 1800 seconds for a hot start or until the solution converges, whichever occurs first. The subroutine begins to check convergence criteria after 1200 seconds model time has elapsed for a cold start or 600 seconds for a hot start. The convergence criteria is then checked after every application of the nine-point filter (after every seventh time-step if the default value is used). Subroutine MXSOLV begins the convergence check by calculating the relative variance of the change in mixing layer depth $V_R\left(\Delta H_m\right)$ over the solution matrix according to the expression

$$V_R \; \Delta H_m = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} \left(H'_{m;i,j} - H_{m;i,j} \right)^2 - n \cdot m \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \left(H'_{m;i,j} - H_{m;i,j} \right) \right]^2 \right\} \Big/ \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} \left(H'_{m;i,j} \right)^2 - n \cdot m \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \left(H'_{m;i,j} \right) \right]^2 \right\} \qquad (4\text{-}7)$$

where

H'_m = layer depth calculated for the present time step at each point in the n by m matrix

H_m = layer depth for the previous time step at each point in the n by m matrix

The subroutine determines the slope of the curve describing $V_R\left(\Delta H_m\right)$ as a function of time and the difference between $V_R\left(H_m\right)$ at the present and previous application of the filter. The solution is assumed to have converged if the

- Slope of the curve describing $V_R(\Delta H_m)$ as a function of time is equal to or less than zero

- Present value of $V_R(\Delta H_m)$ is less than the previous value and the difference between the values is less than or equal to 1×10^{-5}

The program saves the converged mesoscale wind field solution and other parameters calculated in the mixing-layer routines by passing them to the EPAMS data base. At this point the Analysis Routine is terminated and control is returned to XECAMS.

4.2 APPLICATION PHASE OF EPAMS

Inspection of Figure 4-1 above shows the foreman TRYDIF of the Applications Phase of EPAMS is comprised of the two operational elements TRNPRT and DIFFUZ. These operational elements perform the calculations outlined in the block diagram shown in Figure 4-4 after receiving program control through the foreman TRYDIF from XECAMS and the supervisor APPLY.

The subroutine TRNPRT calculates the trajectory of the cloud centroid downwind from the source location using the output from the mesoscale wind-field model and the procedures outlined in Section 2.2.1 above. The program calculates a point on the trajectory at intervals equal to one-tenth the spacing of the solution matrix of the mesoscale model (500 meters in the present case). At each of the calculated points on the trajectory, the program also determines the appropriate mixing-layer depth and wind speed for use in the diffusion models using the interpolation techniques described in Section 2.2.1. After completion of the calculations, program control is passed to subroutine DIFFUZ via TRYDIF.

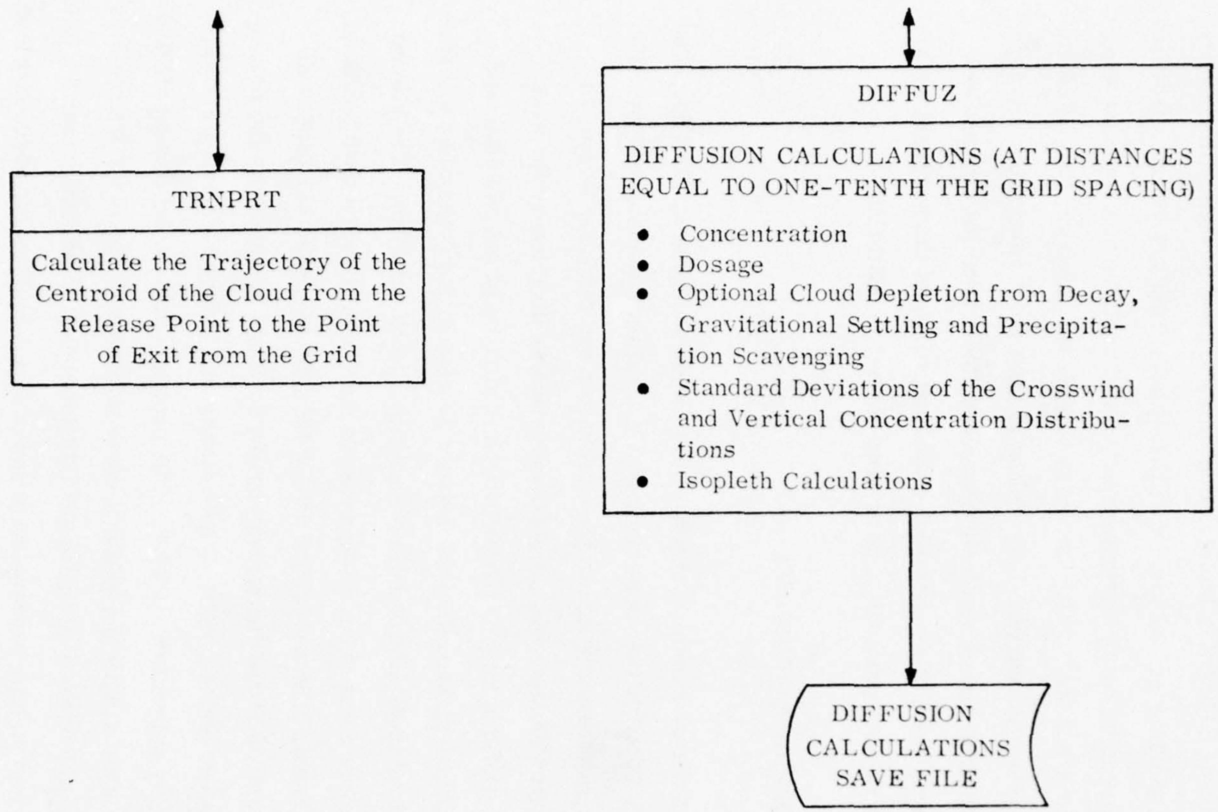


Figure 4-4. Application Phase of EPAMS.

The subroutine DIFFUZ calculates concentration and dosage with optional cloud depletion due to decay and either deposition losses from gravitational settling or precipitation scavenging using the instantaneous and continuous source models described in Section 2.2.2 above. At the users option, the program also calculates concentration and/or dosage isopleth half-widths at each point along the trajectory for instantaneous sources and concentration isopleth half-widths for continuous sources. At the completion of the diffusion model calculations the program writes the solutions to the EPAMS print file and returns control to XECAMS.

4.3 GRAPHICS PHASE OF EPAMS

Figure 4-5 shows a block diagram of the Graphics Phase of EPAMS. The program routine PLTPRG is the supervisor that controls all graphics output. The program retrieves solutions created in the application and Analysis Phases of EPAMS and prints and/or plots the solutions on option as directed by the user through XECAMS. Options include the printing and plotting of intermediate and final wind-field patterns produced by the mesoscale model, isopleths of concentration and dosage, cloud trajectories and mixing-layer depth contours. The plotted solutions are shown on a base map of the solution matrix grid with a background of terrain-height contours. The subroutine labled HEC004 in Figure 4-5 plots and labels the axes of each plot and the terrain height contours are added to the plot by subroutines HEC005 and HEC009. Wind velocity and direction vectors are calculated and plotted by subroutine HEC006. The program can, in subroutine HEC008, recalculate (at the user's option) additional cloud trajectories for other source locations using the solution wind-field in the EPAMS data base. The supervisor PLTPRG can also calculate and display concentration and dosage isopleths for additional levels at the user's option. This feature is useful if the initially requested levels do not show details in which the user is interested or if the initial levels were either so large or so small that they do not occur in the solution matrix. It should be noted

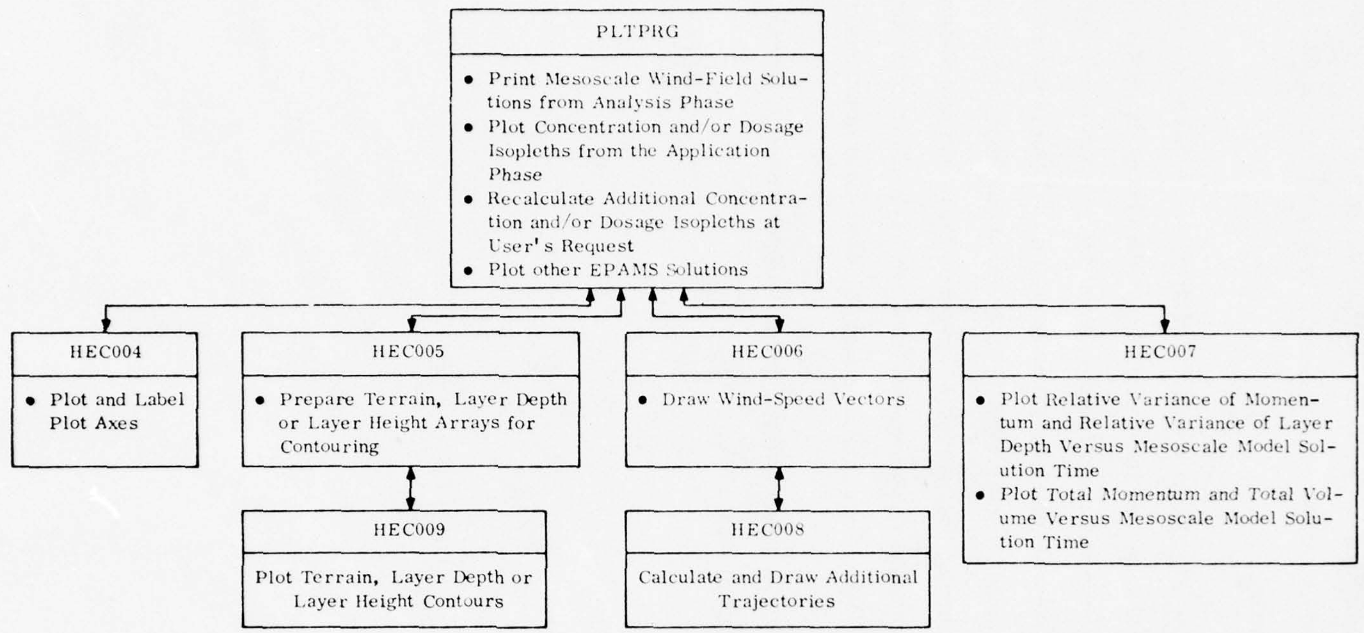


Figure 4-5. Operational elements of PLTPRG Graphics Phase.

that PLTPRG or its subroutines do not recalculate the wind-field nor the diffusion parameters, but use information from the last solutions to recalculate additional trajectories and isopleths.

The subroutine HEC007 controls the plotting of the relative variance of the change in layer depth, the relative variance of the change in momentum, the total mass and total momentum as a function of model solution time. As noted in Section 4.1, the relative variance of the change in layer depth is used as criteria for establishing that the mesoscale wind-field solution has converged. Since the mesoscale wind-field model theoretically conserves mass and momentum, the plots of total mass and momentum versus model time are useful in monitoring the effects of numerical approximations in calculations of the wind-field solutions.

At the completion of the plotting and print operations, program control is returned from PLTPRG to XECAMS.

SECTION 5

EXAMPLE CALCULATIONS

The Analysis and Application Phases of EPAMS were used to produce concentration and dosage isoplith plots for four example cases. The calculations were made for meteorological conditions that occurred in the vicinity of WSMR during the 14th and 15th of November 1974.

The surface weather pattern in the western United States on the 12th of November was characterized by a high-pressure ridge extending from the Pacific Northwest southeastward to Texas. The high-pressure ridge began to break down on the 13th of November with the rapid approach of a low-pressure system moving into the central U. S. from western Canada. The cold front associated with this system tended to remain east of the Rocky Mountains as it pushed south and east during the day. By the morning of the 14th, the low-pressure cell had moved into the northeast and the cold front extended southwestern through the eastern U. S., into Texas and northeast Mexico, then abruptly northward along the eastern slopes of the Rockies. The cold front had actually passed El Paso during the night from the "back-door", or moving from east to west. The passage was not accompanied by significant weather at WSMR, but broken middle and high clouds were reported. The high-pressure system behind the cold front had moved over WSMR by the morning of the 15th of November and calm winds were reported at many surface stations in the area.

The temperature profiles from rawinsonde observations made at El Paso during the period are shown in Figure 4-1. Inspection of Figure 4-1 shown that the cold front had not reached El Paso by 1700 L on the 13th of November. By 0500 L on the 14th, the cold front had passed El Paso and the thermal inversion occurring at about 800 millibars and associated with the frontal surface persisted in the rawinsonde observations made at 1700 L on the 14th and 0500 L on the 15th of November. Surface observations made at El Paso indicate that relatively shallow

surface-based inversions formed during the nighttime and were eliminated by surface heating during the early morning hours.

Three of the example cases were calculated for 0700 L, 0900 L and 1300 L on the 14th of November to illustrate the performance of the Analysis Phase in analyzing and modifying the 0500 L rawinsonde observation made at El Paso on the 14th and the resulting effects on the concentration and dosage isoplith patterns produced by the Applications Phase of EPAMS. The fourth case was calculated for 0300 L, 15 November to illustrate program performance for very stable light-wind conditions at the surface. These calculations were all made for a hypothetical point source located about 31 kilometers southeast of WSMR using the default values for point source parameters given in Appendix A except the source strength Q was set to 100 units. The major terrain features of the solution grid are identified in Figure 4-2. The results of the calculations are described below.

5.1 INSTANTANEOUS POINT SOURCE CALCULATIONS FOR 0700L, 0900L AND 1300L, 14 NOVEMBER 1974

Program requests were made for dosage and concentration calculations downwind from the hypothetical instantaneous point source southeast of WSMR for 0700L, 0900L and 1300L, 14 November 1974.

Because the request for a solution at 0700L was made for a time less than three hours after the 0500L rawinsonde observation at El Paso, the program used the temperature profile for 0500L shown in Figure 5-1 without modification of the surface, 850 millibar and 700 millibar temperatures and winds to calculate inputs to the mesoscale model. The rawinsonde analysis routine (MIXLYR) ignored the surface-based inversion indicated by the 0500L rawinsonde temperature profile because it was less than 100 meters in depth. Since the elevated inversion with a base at 832 millibars meets the criteria set for determining inversion bases described in Section 3.1, the analysis routine defines the mixing depth as 1697 meters MSL or 491 meters above ground-level (AGL). The height-weighted mean wind components determined from the rawinsonde data below this level were u equal to -4.78 m sec^{-1} and v equal 2.30 m sec^{-1} or a wind direction of 116 degrees at 4.8 m sec^{-1}.

The mesoscale wind-field model solution for 0700L is shown in Figure 5-3. Inspection of Figure 5-3 shows that southeasterly winds dominate the flow pattern in the southern portions of the Tularosa Valley and the valley Jornada Del Muerto west of the Organ Mountains. Since the highest elevations of the San Andres, Organ and Sacramento Mountains extend above the depth of the surface mixing layer, they form an effective barrier to the southeasterly flow and divert the wind flow northward in the northern portion of the Tularosa Valley. The mesoscale wind-field model has set the wind components to zero and reduced the mixing layer to 30 meters above terrain height in the higher elevations, which has resulted in down-

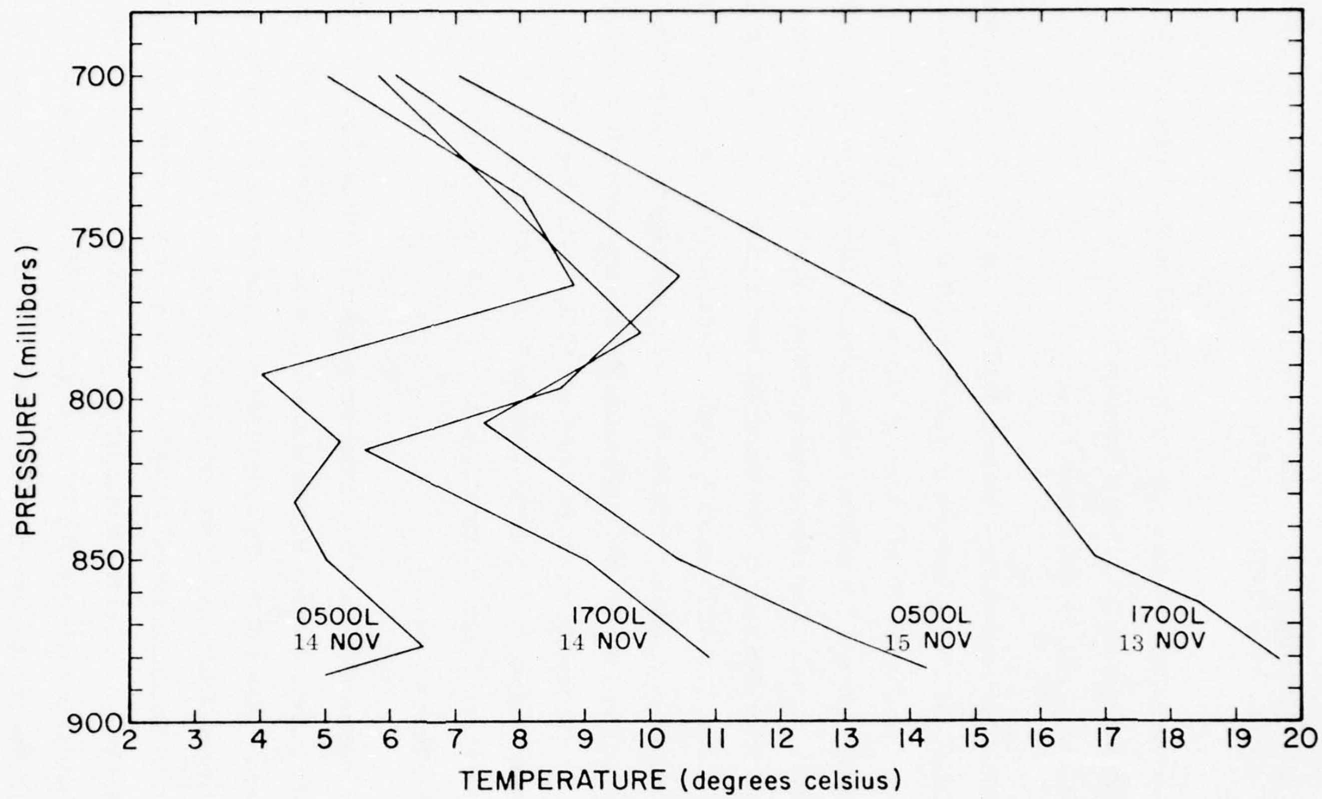


Figure 5-1. Temperature profiles from El Paso, Texas rawinsonde observations for 13, 14 and 15 November 1974.

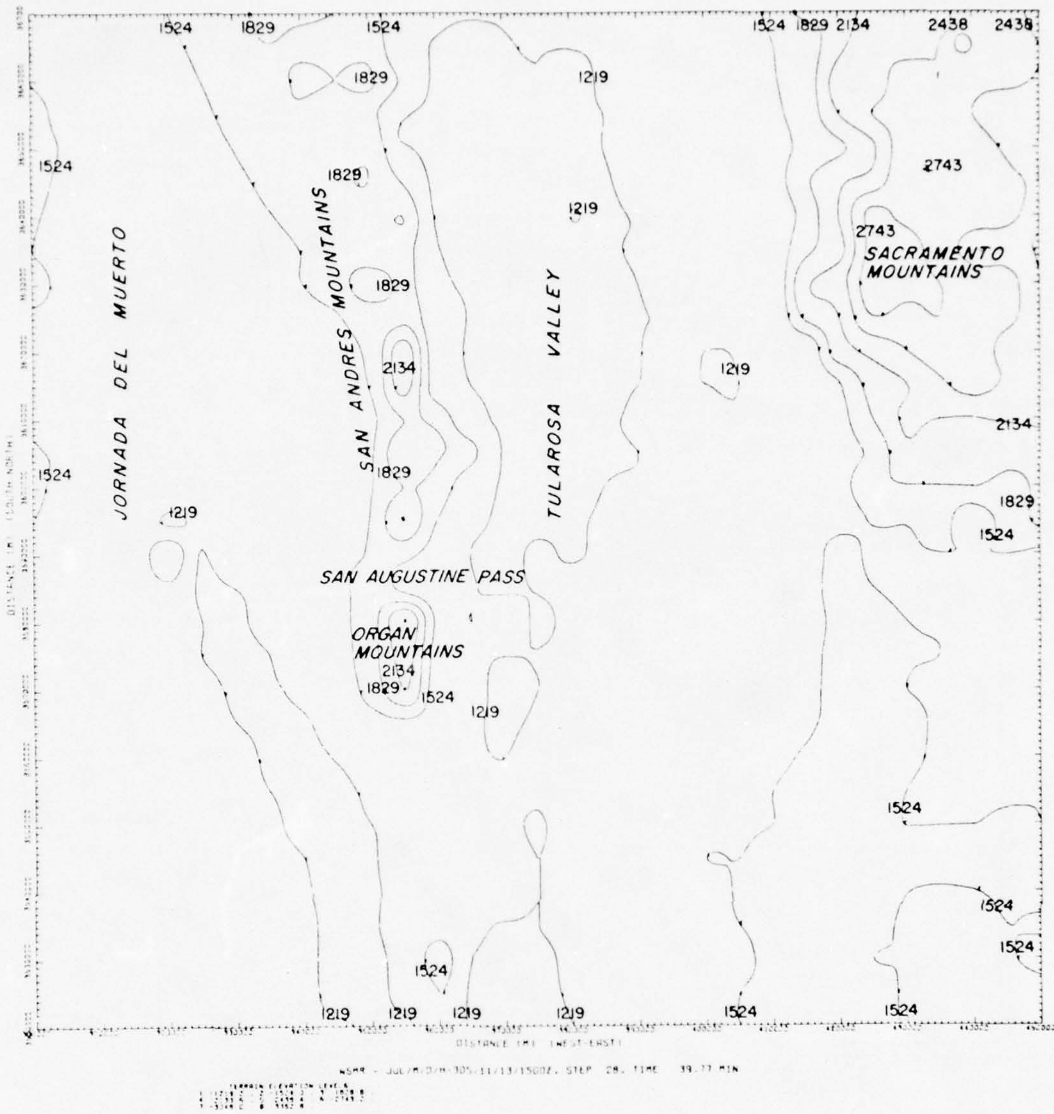


Figure 5-2. Map showing major topographic features of the mesoscale wind-field solution grid.

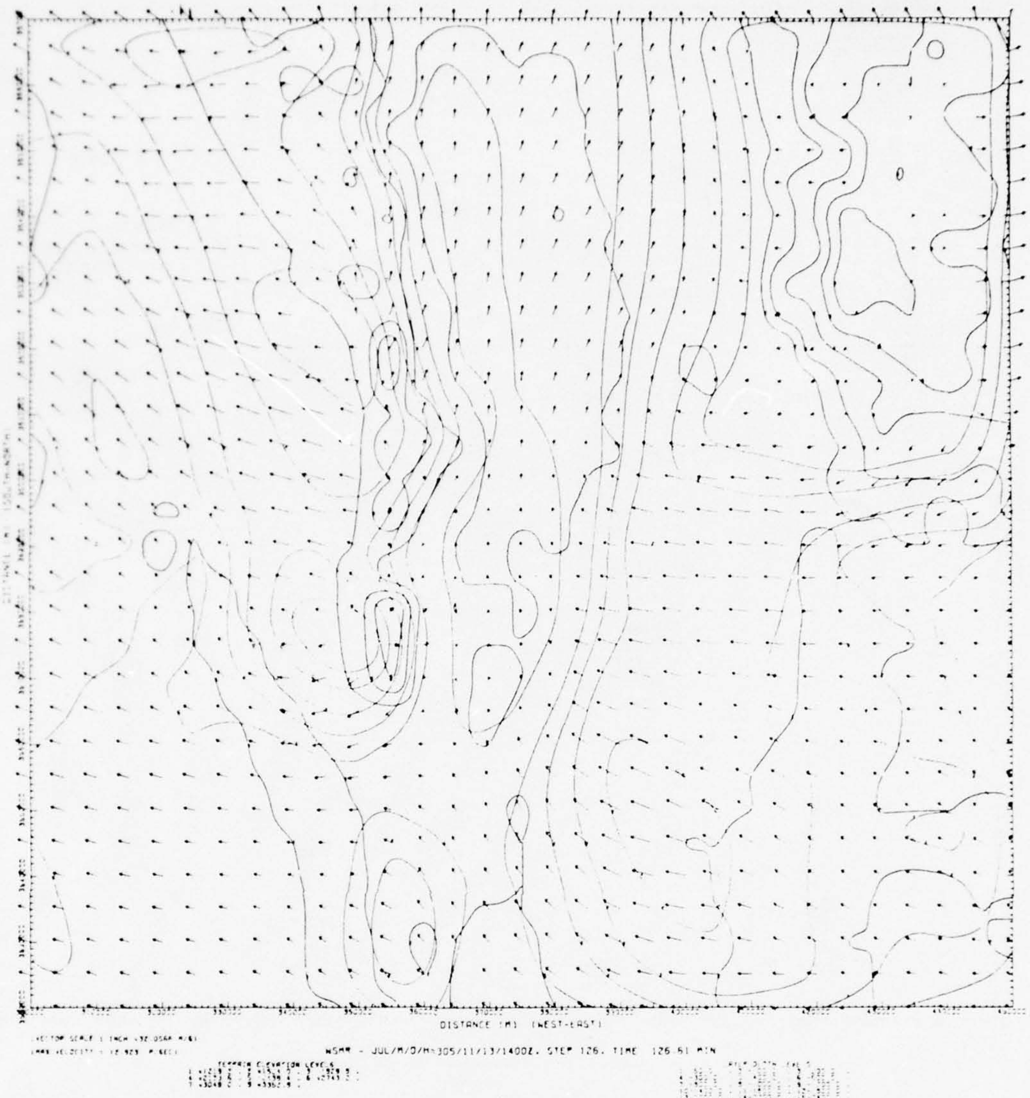


Figure 5-3. Mesoscale wind-field model solution for 0700L, 14 November 1974.

slope flow as evidenced by the flow pattern just east of the Organ Mountains. A typical leeside depression of the mixing depth has formed to the west of the San Andres and Organ Mountains due to hydraulic jump effects and layer thickening has occurred east of the mountains.

The results of the transport/diffusion model calculations for 0700L are shown in Figure 5-4. The cloud trajectory is diverted northward by the downslope flow occurring in the vicinity of the Organ Mountains and follows the south-southwesterly flow through the Tularosa Valley. Concentration isopleths, indicated by the heavier lines in Figure 5-4, are drawn for 1×10^{-4}, 1×10^{-6} and 1×10^{-12} source units per cubic meter. The dosage isopleths indicated by the thinner lines are drawn for 1×10^{-4}, 1×10^{-6} and 1×10^{-12} source units per cubic meter. The isopleths were calculated using the diffusion model parametric inputs described in Section 3.2 associated with a net radiation index of -1 since the hour was nighttime, where nighttime is defined in the program as extending from one hour before sunset to one hour after sunrise, and the cloud cover exceeded five-tenths.

The inputs to the mesoscale wind-field model for the 0900L request for transport/diffusion calculations were still based on the 0500L rawinsonde measurement at El Paso, but the rawinsonde observation was modified because the observation was more than three-hours old at 0900L. The modifications made by the mixing-layer analysis routine are shown in Figure 5-5. The routine has inserted the 0900L surface temperature and winds from El Paso and the GWC predicted 700 millibar temperature and winds into the rawinsonde observation and has removed the 0500L temperature and winds observed at the surface and 812, 832, 877 and 700 millibar levels. As a result, the analysis routine calculates a mixing depth of 2099 meters MSL (893 meters AGL) based on the increase in virtual temperature which occurs at 792 millibars. The mean wind components calculated for the mixing layer were u equal -4.87 m sec^{-1} and v equal 1.89 m sec^{-1}, or a wind direction of 111 degrees at 5.2 m sec^{-1}.

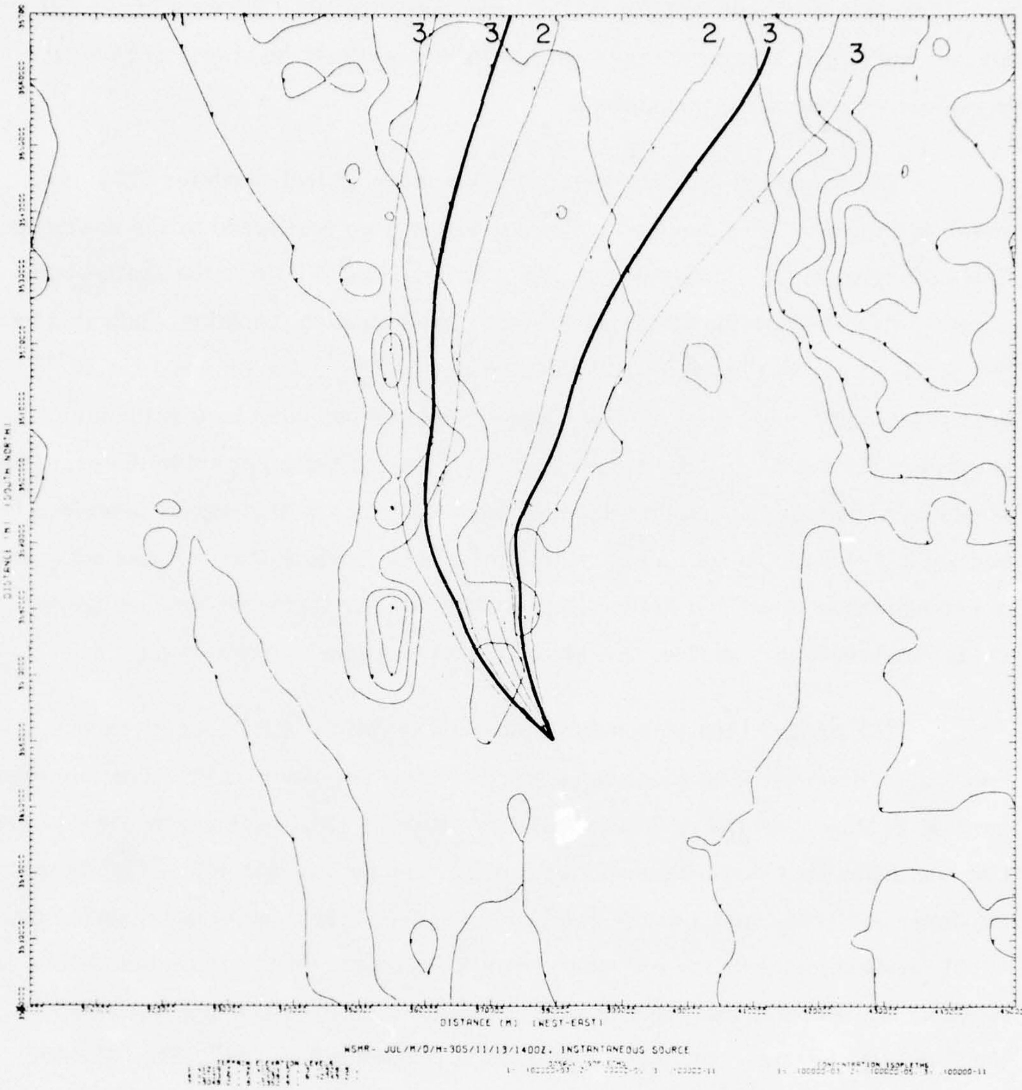


Figure 5-4. Concentration isopleth (heavy line) and dosage isopleths for 0700L, 14 November 1974. Codes for isopleth levels are identified just below the computer plot.

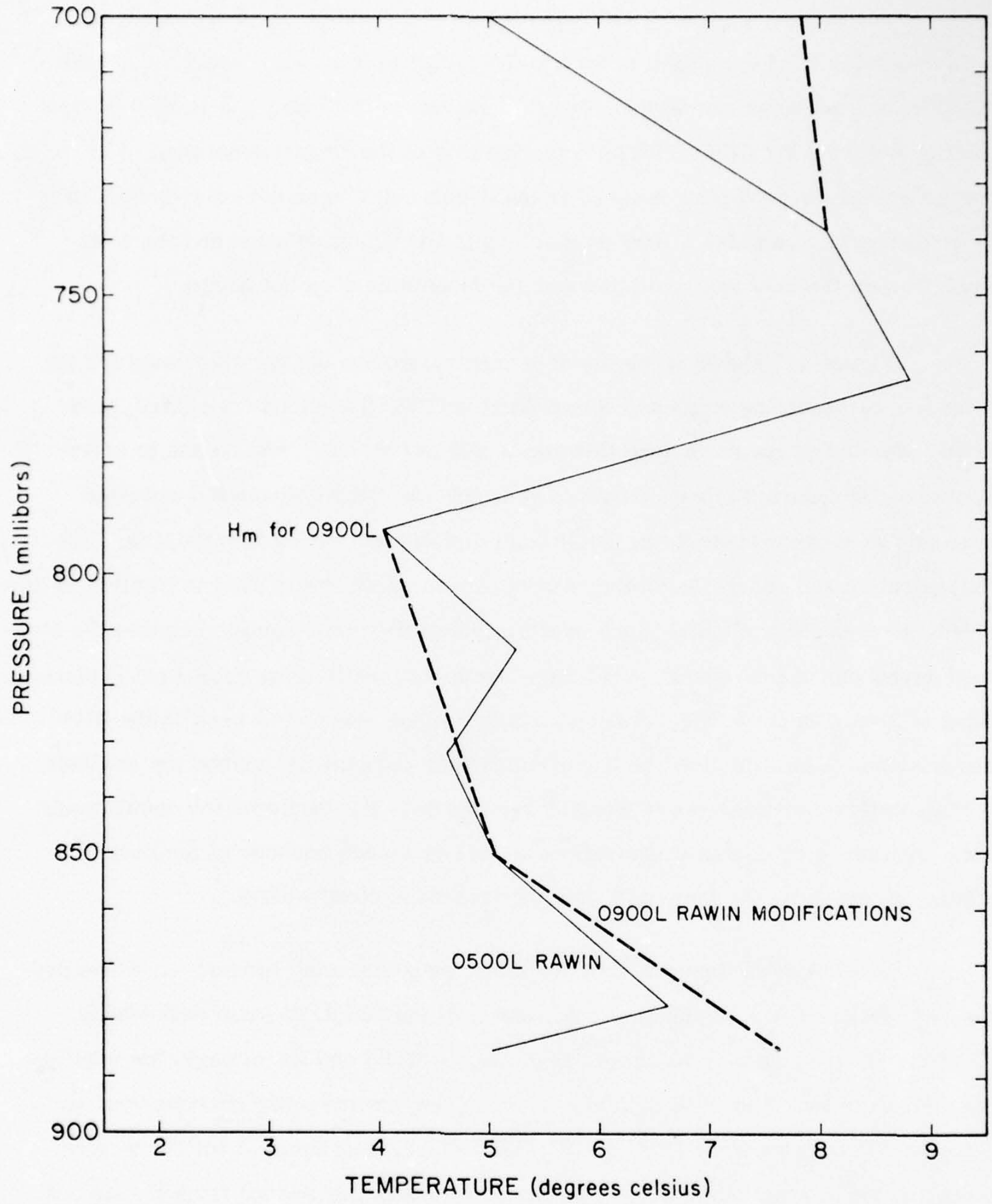


Figure 5-5. El Paso rawinsonde observation at 0500L, 14 November 1974 (solid line) and modifications made by the mixing-layer analysis routine for the 0900L request for a transport/diffusion calculation (broken line).

The mesoscale wind-field model solution for 0900L is shown in Figure 5-6. Since the wind components input to the model at 0900L are nearly identical to those used as inputs for 0700L, the solution shown in Figure 5-6 resembles the solution obtained for 0700L except in the vicinity of the Organ Mountains. The deeper mixing-layer depths obtained in the 0900L solution do not restrict the flow over the Organ Mountains nearly as much as in the 0700L solution and the downslope flow on the east and south sides of the mountains does not occur.

Figure 5-7 shows isopleths of concentration and dosage calculated for the point source assuming a release occurred at 0900L. The cloud trajectory is no longer affected by the downslope flow that occurred at 0700L and begins to travel westward, is diverted slightly southward around the Organ Mountains and then proceeds west-northwesterly in the Jornada Del Muerto valley flow pattern. The concentration and dosage isopleths, drawn for the same levels used in Figure 5-4 for 0700L, have become slightly more broad in the crosswind direction because the standard deviations of the azimuth wind angle are significantly larger for a net radiation index of 2 used in the 0900L calculations than for the index of -1 used in the 0700L calculations. However, had the calculations been carried-out beyond the confines of the solution matrix shown in Figures 5-4 and 5-7 the isopleths for 0900L would have terminated or closed at shorter downwind distances because of the deeper mixing depths along the trajectory and the increased cloud widths.

The latest rawinsonde data available for performing concentration and dosage calculations for a release at 1300L was still the 0500L El Paso rawinsonde. Since the rawinsonde data was more than six hours old and the criteria for modifying the rawinsonde and GWC data was not met, the mixing-layer analysis routine selected a default value of 1997 m MSL (792 AGL) from Table 3-1 for the mixing depth and the surface wind of 141 degrees at 3 meters per second from the surface observations at El Paso for 1300L. The wind-flow pattern for 1300L produced by the mesoscale model, shown in Figure 5-8, indicates southeasterly flow over the

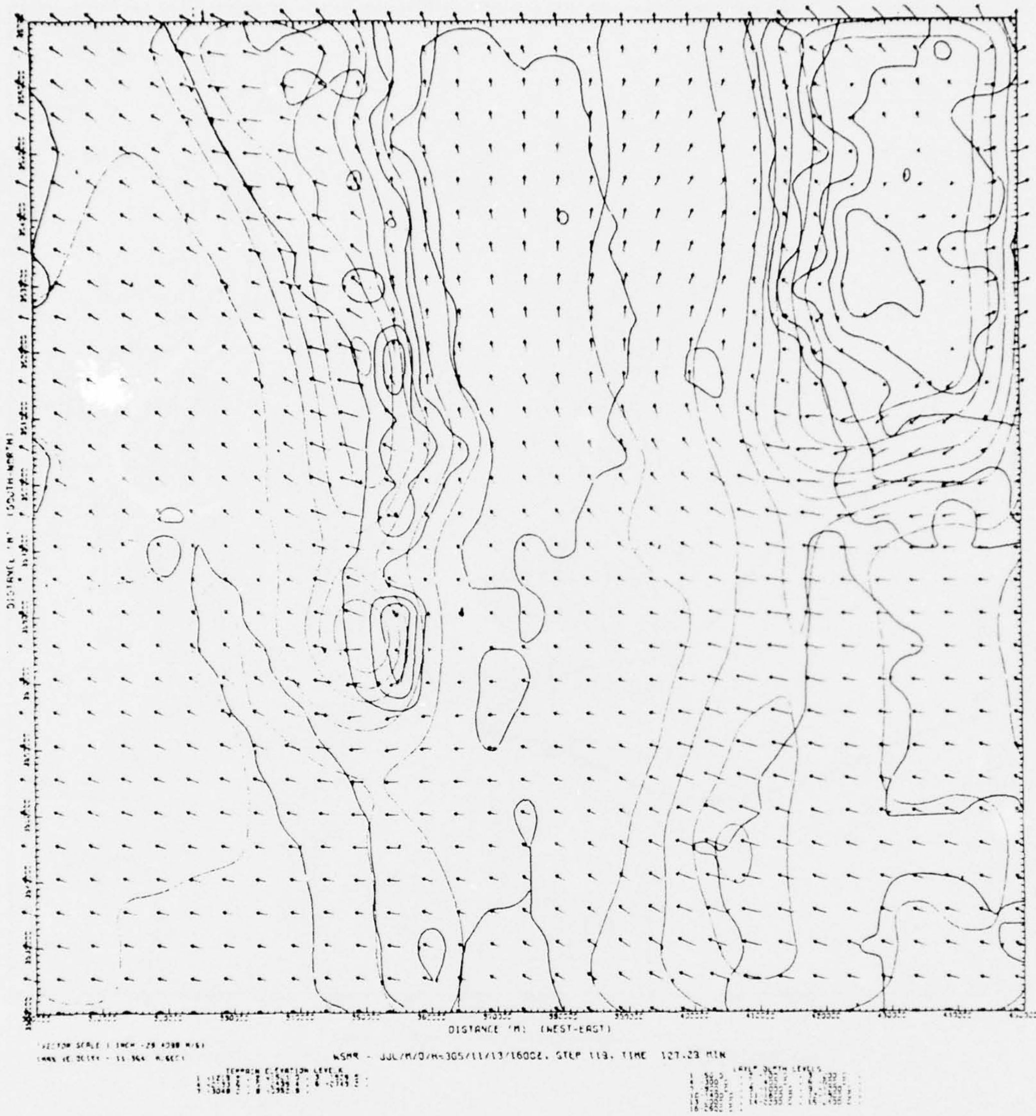


Figure 5-6. Mesoscale wind-field model solution for 0900L, 14 November 1974.

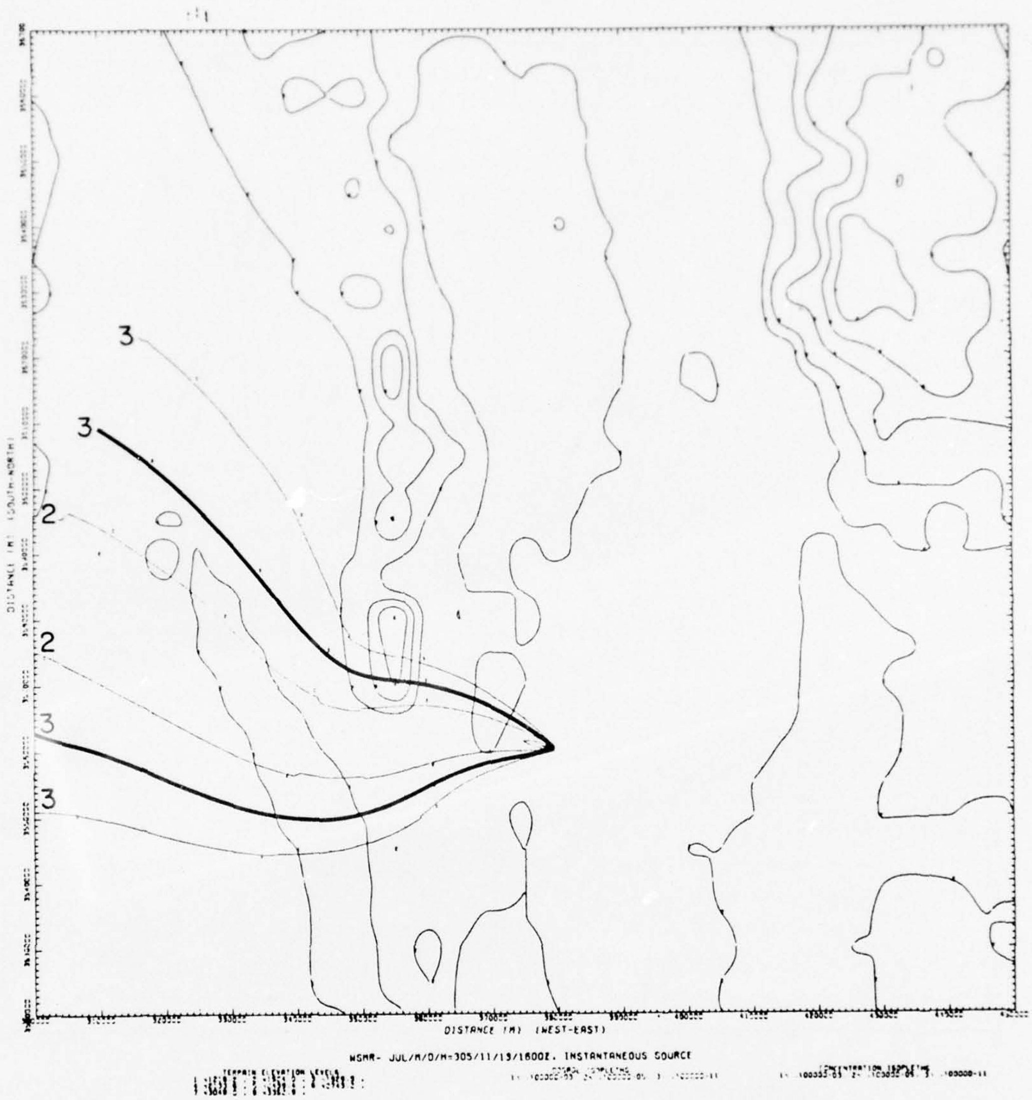


Figure 5-7. Concentration isopleth (heavy line) and dosage isopleths for 0900L, 14 November 1974. Codes for isopleth levels are identified just below the computer plot.

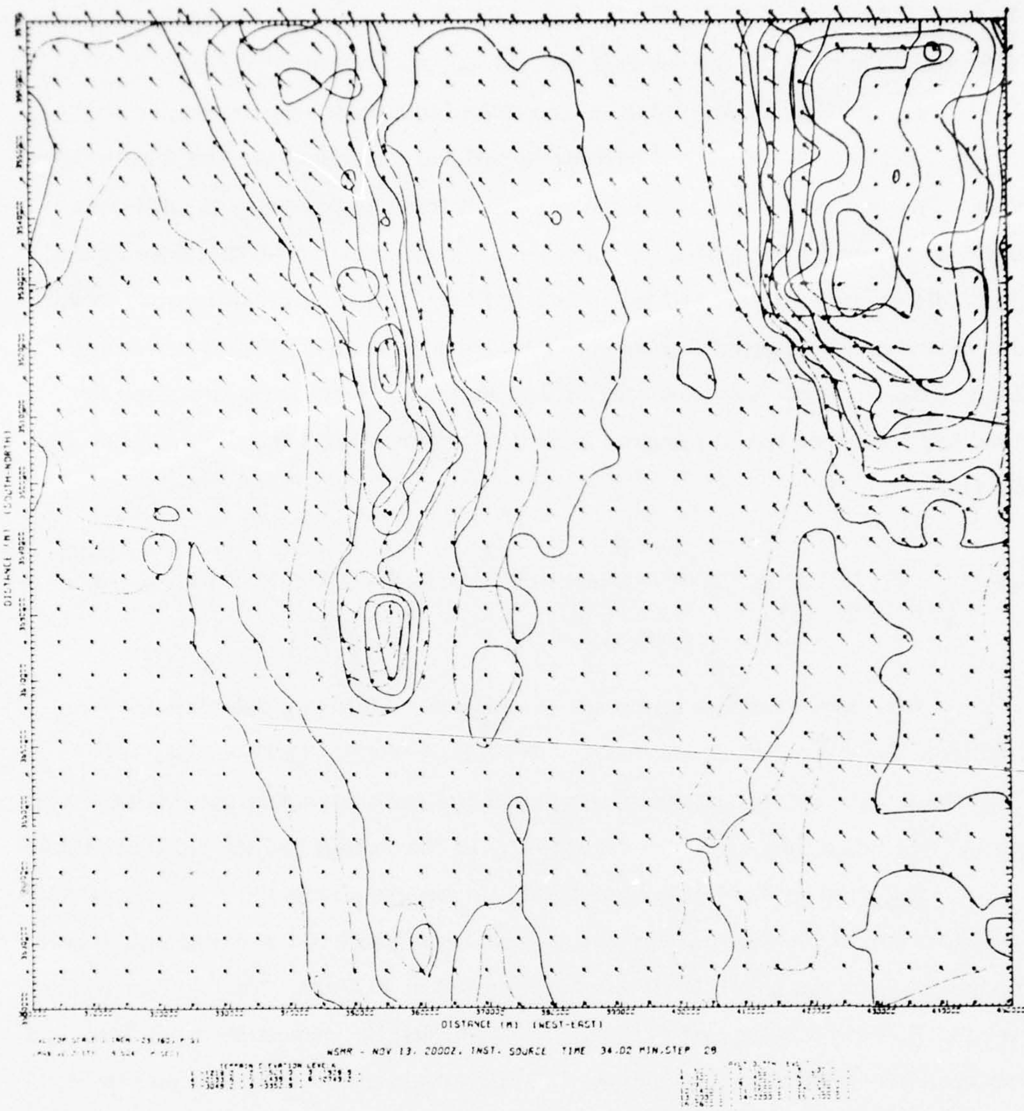


Figure 5- 8. Mesoscale wind-field model solution for 1300L, 14 November 1974.

solution grid except immediately upwind of the San Andres and Organ Mountains and in the vicinity of the Sacramento Mountains. As indicated in Figure 5-9 showing the concentration and dosage isopleths for a point source release at 1300L, the cloud trajectory is directed northwestward and intersects the San Andres Mountains. The isopleths begin to expand rapidly at this point because the diffusion model compensates for attempted increases in concentration (in this case caused by rapidly decreasing mixing layer depth and insufficient increase in wind speed to compensate for reduced mixing depth) by increasing cloud dimensions in the plane of the horizon. The isopleths expand in a more regular fashion after the cloud has traversed the San Andres Mountains and reached the valley Jornado De Muerto.

5.2 INSTANTANEOUS POINT SOURCE CALCULATIONS FOR 0300L, 15 NOVEMBER 1974

A program request was made for transport/diffusion calculations downwind from the point-source southeast of WSMR for 0300L, 15 November 1974. Since the latest rawinsonde observation available for analysis by the mixing-layer routine was more than six hours old (1700L, 14 November) and the GWC prediction data was rejected on temperature criteria, the program selected a default value of 30 meters for the surface mixing-layer depth from Table 3-1 and a default wind component from the 0300L surface wind reported at El Paso. The surface wind speed at El Paso was reported as calm at 0300L and the mesoscale wind field model was initialized with zero momentum. The flow-field produced by the mesoscale model is shown in Figure 5-10. As indicated in the figure, downslope flow is occurring everywhere, but is greater west of the Sacramento Mountains and in the region just east of San Augustine Pass separating the Organ and San Andres Mountains. The cloud, as indicated in Figure 5-11 by the isopleth patterns, begins to move southwestward for a few kilometers, then moves toward the low elevations in the Tularosa

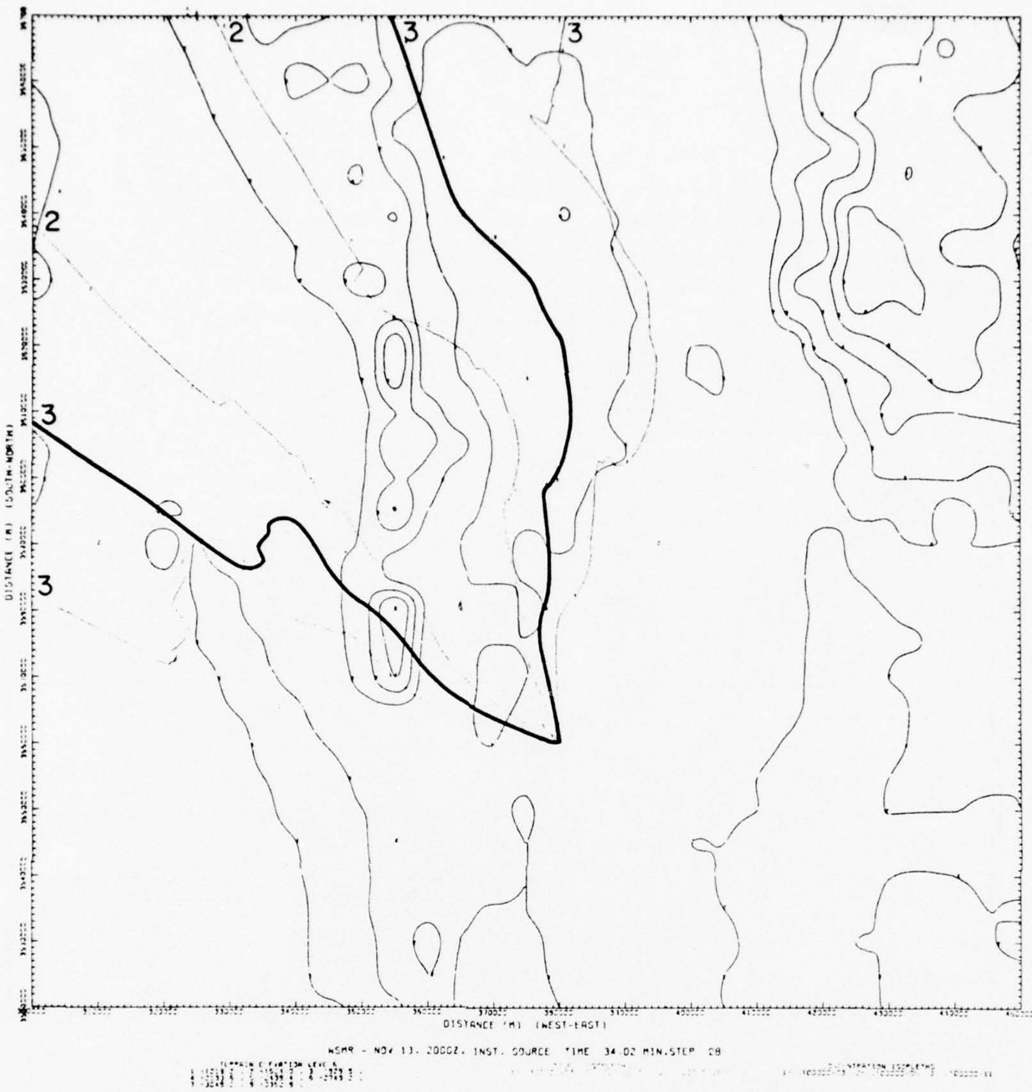


Figure 5-9. Concentration isopleth (heavy line) and dosage isopleths for 1300L, 14 November 1974. Codes for isopleth levels are identified just below the computer plot.

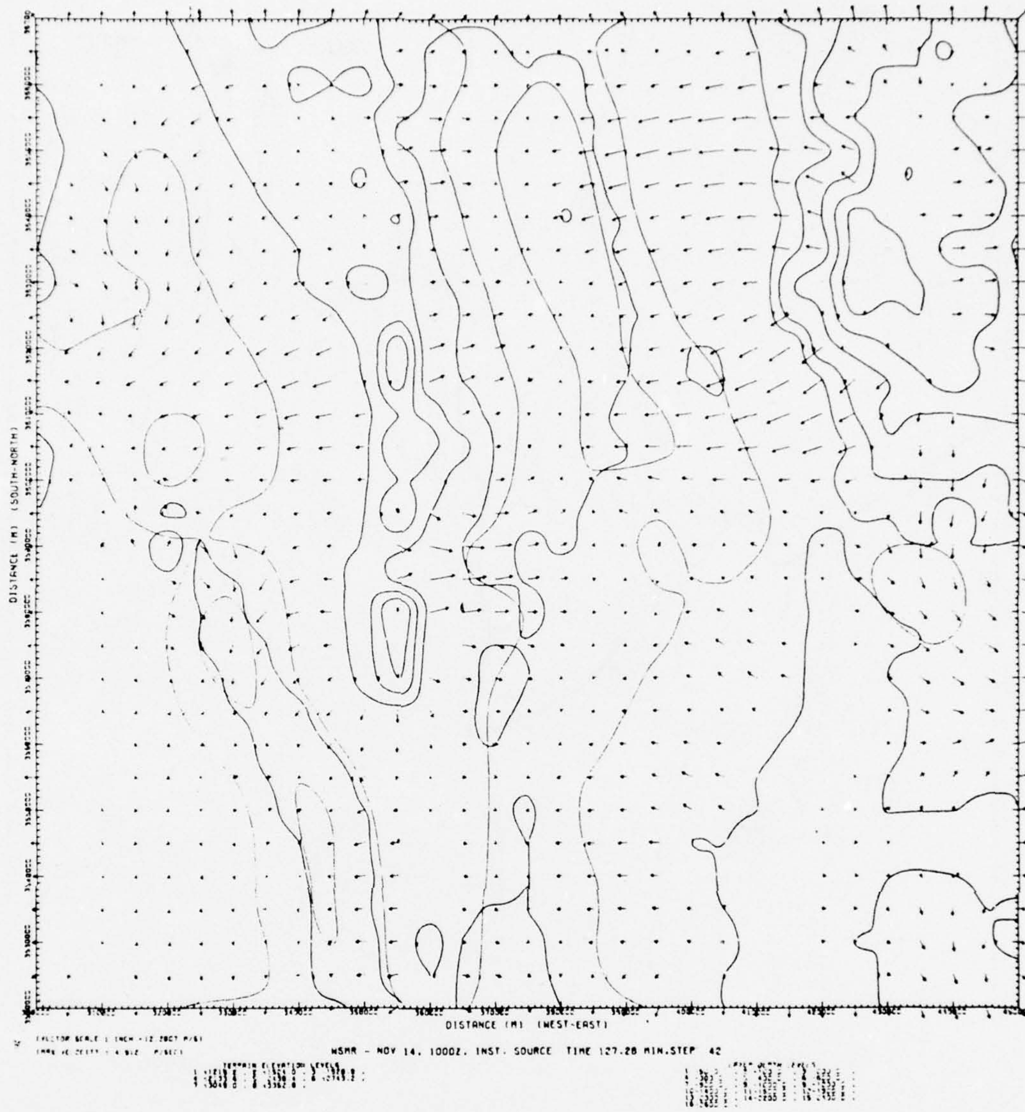


Figure 5-10. Mesoscale wind-field model solution for 0300L, 15 November 1974.

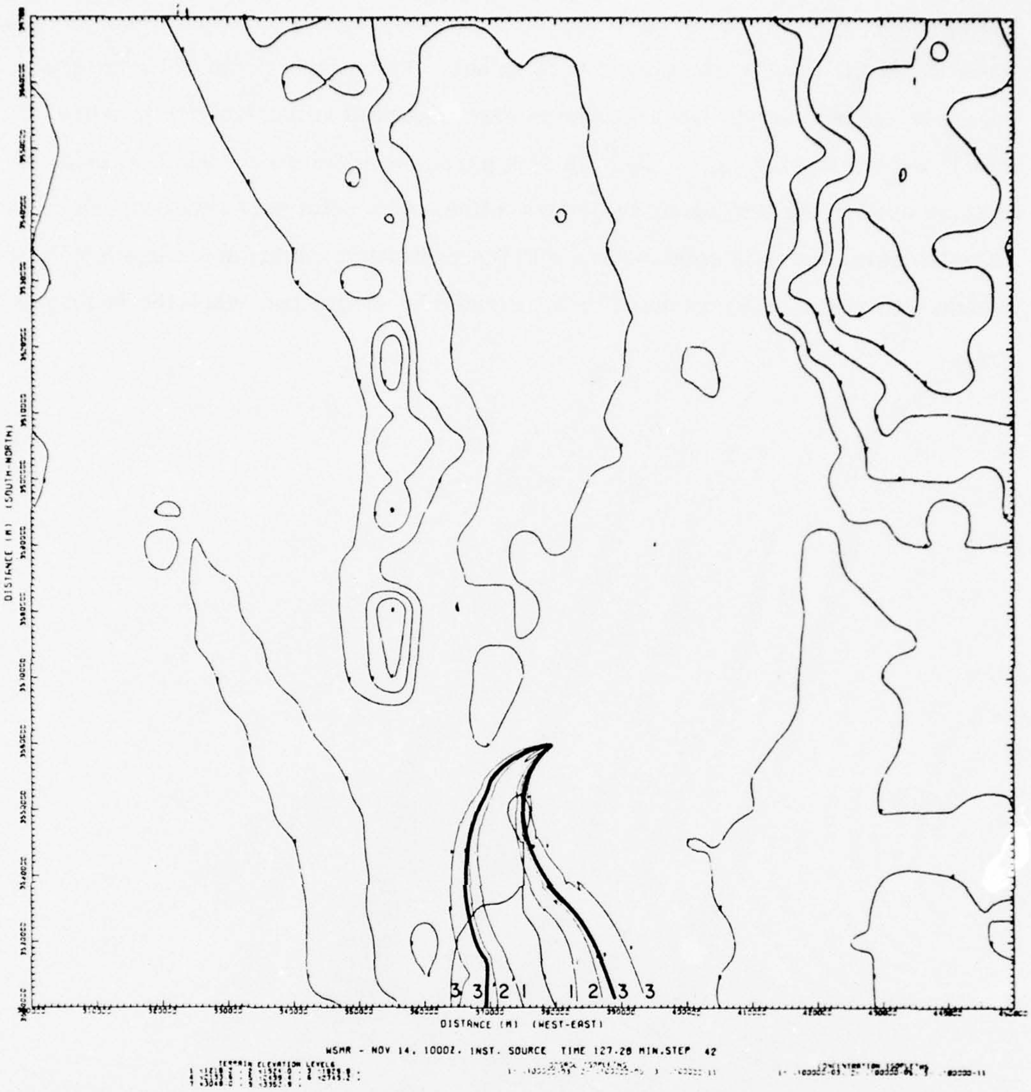


Figure 5-11. Concentration isopleth (heavy line) and dosage isopleths for 0300L, 15 November 1974. Codes for isopleth levels are identified just below the computer plot.

Valley at the southern boundary of the solution matrix grid. The highest wind speed along the cloud trajectory of 0.86 m sec^{-1} and mixing depth of 79 meters occurs at the boundary. Because the release occurred at nighttime with more than five-tenths cloud cover, the diffusion parameters for a net radiation index of -1 were used in the transport/diffusion routine. The relatively small values of the diffusion parameters in combination with the restricted mixing depth and low wind speeds lead to the high concentrations indicated by the narrow isopleths in Figure 5-11.

REFERENCES

Briggs, G. A., 1971: Some recent analyses of plume rise observations. In Proceedings of the Second International Clean Air Congress, Academic Press, New York.

Briggs, G. A., 1972: Chimney plumes in neutral and stable surroundings. Atm. Env., 6(7), 507-510.

Carnahan, Luther, and Wilkes, 1969: Applied Numerical Methods. Wiley and Sons, Inc., New York, New York.

Cramer, H. E., et al., 1972: Development of dosage models and concepts. Technical Report, Contract No. DAAD09-67-6-6-0020(R), Desert Test Center, Fort Douglas, Utah.

Dumbauld, R. K., et al., 1970: Handbook for estimating toxic fuel hazards. Final Report under Contract No. NAS8-21453, NASA Report No. CR-61326.

Dumbauld, R. K., J. R. Bjorklund and J. F. Bowers, 1973: NASA/MSFC multilayer diffusion models and computer program for operational prediction of toxic fuel hazards. NASA Contractor Report NASA CR-129006. H. E. Cramer Company Technical Report prepared for National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Alabama 35812.

Holzworth, G. C., 1967: Estimates of mean maximum mixing depths in the contiguous United States, Mon. Wea. Rev., 92, 235-242.

Lavoie, R. L., 1972: A mesoscale numerical model of lake-effect storms. J. Atm. Sci., 29(6), 1025-1040.

Norton, C. and G. Hoidale, 1975: The diurinal variation of mixing depth by month over White Sands Missile Range, N. M. Research and Development Technical Report ECOM-5579, Atmospheric Sciences Laboratory, U. S. Army Electronics Command, White Sands Missile Range, New Mexico 88002.

Pasquill, F., 1962: Atmospheric Diffusion. D. Van Nostrand Co., Ltd., London, 297.

Shapiro, R., 1970: Smoothing, filtering and boundary effects. Reviews of Geophysics and Space Physics, 8(2), 359-387.

Swanson, R. N. and M. M. Hoidale, 1962: Low-level wind profile prediction techniques. Progress Report 4, SMSA, White Sands Missile Range, New Mexico.

Swanson, R. N. and H. E. Cramer, 1965: A study of lateral and longitudinal intensities of turbulence. J. Appl. Met., 4, 409-417.

Tingle, A. G. and J. R. Bjorklund, 1973: Study and investigation of computer algorithms for the solution of the shallow-fluid equations as a means of computing terrain influences on wind fields. H. E. Cramer Company Technical Report TR-73-302-01 prepared under Contract No. DAAD07-73-C-0309 with ASL, ECOM, White Sands Missile Range, New Mexico.

Turner, D. B., 1964: A diffusion model for an urban area. J. Appl. Meteor., 3(1), 83-91

Woolf, H. M., 1968: On the computation of solar elevation angles and the determination of sunrise and sunset times. NASA TMX-1646. National Meteorological Center, ESSA, Hillcrest Heights, Maryland.

APPENDIX A

USERS' INSTRUCTIONS FOR THE ANALYSIS, APPLICATION AND GRAPHICS PHASES OF EPAMS

This appendix gives users' instructions for the three main phases of EPAMS. The computer programs that comprise the many phases of EPAMS are written in FORTRAN V and are designed for use on a Univac 1108 computer. A complete listing of the EPAMS computer program is given in Appendix B. Also, a block diagram of the main components of each phase of EPAMS, as well as, temporary executive routines is given in Section 4 of the main body of this report. The program has been delivered to WSMR with three dummy executive routines, these routines are shown as XECAMS, ANALIZ and APPLY in Figure 4-1, Section 4. The final design of these routines is not the responsibility of the H. E. Cramer Company and as such they are only presented here for completness and to provide temporary drivers for the Analysis, Application and Graphics Phases. See Section A. 7 below for data file numbers assigned by the temporary executive XECAMS.

The input variables required in the temporary executive XECAMS are described in Section A. 1 below. Also, optional input variables for the Analysis, Application and Graphics Phases are given in Sections A. 2, A. 3, and A. 4, respectively. An example execution card deck is given in Section A. 5 and the input data method used in the Analysis, Application and Graphics Phases is given in Section A. 6

A. 1 EPAMS EXECUTIVE INPUTS

The temporary executive XECAMS requires data input via the EPAMS input message file to direct the executive to execute one of the three phases of EPAMS. The set of variables given below are read by XECAMS prior to the execution of any of the EPAMS phases. One data card is read for each phase and only those variables

required by the particular phase need be input. A summary table of the input variables and their respective units and default values is given in Table A-1.

FORTRAN VARIABLE	FORMAT	CARD COLUMNS	DEFINITION
TYPE 1	I3	1-3	This integer variable is used to direct the executive to execute one of the EPAMS Phases. If set equal to 2, the Analysis Phase is executed and XECAMS calls the Analysis Supervisor ANALIZ. If set equal to 1, the Application Phase is executed and XECAMS calls the Application Supervisor APPLY. If set equal to 4 the Graphics Phase is executed and XECAMS calls the Graphics Supervisor PLTPRG. (This variable is referenced only in the executive XECAMS.)
TYPE 2	I3	4-6	This integer variable is used to execute alternate sections of the Analysis and Application Phases. If set equal to 1 or 2 and TYPE 1 equals 2, the mixing layer and mesoscale wind field programs of the Analysis Phase are executed by a call to the mixing layer and mesoscale wind field programs. If set equal to 1 and TYPE 1 equals 1, the transport/diffusion programs of the

TABLE A-1

SUMMARY TABLE OF INPUT VARIABLES
FOR THE EPAMS TEMPORARY
EXECUTIVE XECAMS

FORTRAN VARIABLE	UNITS	FORMAT	CARD COLUMNS	LIMITS	DEFAULT VALUE
TYPE1	N/A	I3	1-3	= 1, 2 or 4	N/A
TYPE 2	N/A	I3	4-6	= 1 or 2	N/A
TYPE3	N/A	I3	7-9	$\geq$-9 and $\leq$9	0
JULIAN	Julian Day	I3	10-12	= end of last month	N/A
MØNTH	Month	I3	13-15	$\geq$1 and $\leq$12	N/A
DAY	Day	I3	16-18	$\geq$1 and $\leq$31	N/A
HØUR	Hour (GMT)	I3	19-21	$\geq$00 and $\leq$23	N/A
MINUTE	Minute	I3	22-24	$\geq$00 and $\leq$60	N/A
XPLACE	Meters	F10.2	25-34	$\geq$300000 and $\leq$450000	360000
YPLACE	Meters	F10.2	35-44	$\geq$3520000 and $\leq$3670000	3585000
ZPLACE	Meters	F10.2	45-54	$\geq$1310	1310

AD-A038 399 CRAMER (H E) CO INC SALT LAKE CITY UTAH F/G 4/2
MIXING-LAYER ANALYSIS ROUTINE AND TRANSPORT/DIFFUSION APPLICATI--ETC(U)
MAR 77 R K DUMBAULD, J R BJORKLUND DAAD07-76-C-0023
UNCLASSIFIED TR-76-106-01 ECOM-77-2 NL

2 OF 3
AD
A038399

FORTRAN VARIABLE	FORMAT	CARD COLUMNS	DEFINITION
TYPE 2 (Continued)			Application Phase are executed by a call to program TRYDIF. TYPE 2 is not referenced in the Graphics Phase.
TYPE 3	I3	7-9	This integer variable is used to control input data to parts of the Analysis, Application and Graphics Phases. If TYPE 3 is set less than or equal to zero, no input data is read by the particular phase being executed. The programs will default or calculate all potential input data variables. Specifically if TYPE 3 is positive (greater than zero) and the mixing layer/mesoscale section of the Analysis Phase is being executed, the data variables given in Section A. 2 are input via EPAMS file XINFIL (data card input). If the absolute value of TYPE 3 is equal to 9, the program only executes the mixing layer solution. This option is used only to display meteorological data and no solution files are generated or altered. If TYPE 3 is positive (greater than zero), and the transport/diffusion

FORTRAN VARIABLE	FORMAT	CARD COLUMNS	DEFINITION
TYPE3 (Continued)			section of the Application Phase is being executed, the data variables given in Section A.3 are input via EPAMS file XINFIL (data card). If TYPE3 is zero, the program defaults to an instantaneous source with no input data. If equal to -1 the program defaults to a continuous SO_2 source given by the ASARCO copper stack in El Paso, Texas. If equal to -2, the program defaults to a continuous SO_2 source given by the ASARCO lead stack in El Paso, Texas. If TYPE3 is positive (greater than zero) and the Graphics Phase is being executed, the variables given in Section A.4 are input via EPAMS file XINFIL (data card). If any of the three phases of EPAMS cannot complete its task, the particular phase will set TYPE3 equal to a negative number and return control to the executive routines. If the phase solution was completed TYPE3 is greater than or equal to zero upon return.

FORTRAN VARIABLE	FORMAT	CARD COLUMNS	DEFINITION
JULIAN	I3	10-12	The Julian day of the last day of the month prior to the month of the desired solution date and time. (Referenced only in Analysis Phase.)
MØNTH	I3	13-15	Month of the desired solution. (Referenced only in Analysis Phase.)
DAY	I3	16-18	Day of the desired solution. (Referenced only in Analysis Phase.)
HØUR	I3	19-21	Hour of the desired solution (past, present or future) in Greenich mean time (00 to 23). (Referenced only in Analysis Phase.)
MINUTE	I3	22-24	Minute of the desired solution (00 to 60). (Referenced only in the Analysis Phase.)
XPLACE	F10.2	25-34	The UTM (Universal Transvers Mercator) easterly (x) coordinate of the source location in meters. If equal to blank or zero, the program uses 360000 meters.
YPLACE	F10.2	35-44	The UTM northerly (y) coordinate of the source location in meters. If equal to blank or zero, the program uses 3585000 meters.

FORTRAN VARIABLE	FORMAT	CARD COLUMNS	DEFINITION
ZPLACE	F10.2	45-54	The elevation (MSL) of the source location in meters. If equal to blank or zero, the program uses 1310 meters.

A.2 ANALYSIS PHASE INPUTS

This section describes the variables input to the mixing layer part of the Analysis Phase. The variables given below are read only if the variable TYPE3 from the executive is positive (greater than zero). These variables are read in the FORTRAN namelist format through namelist QLST1 via EPAMS file XINFIL (data card). The namelist format is explained in Section A.6 below. A summary table of the input variables is given with their respective units and default values in Table A-2. Possible input variables are:

FORTRAN VARIABLE	DEFINITION
UN	Vector component u of the mean layer wind for use in the mesoscale wind field calculation in units of meters/second. The component u is positive toward the east and negative toward the west. (Default = calculated from the meteorological data base.)

$$UN = V \cdot \sin(\theta \pm 180.0)$$

where V = mean layer wind speed

θ = mean layer wind direction measured from 0 degrees north clockwise (direction from which wind is blowing)

TABLE A-2

SUMMARY TABLE OF THE INPUT VARIABLES FOR THE ANALYSIS PHASE INPUT THROUGH NAMELIST QLST1

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE①	INITIAL VALUE
UN	Meters/ Second	>-99.0	②	-99
VN	Meters/ Second	>-99.0	②	-99
PN	Meters (MSL)	$\geq$1340	②	-99
GN	Meters/ $Second^2$	>-99.0	②	-99
ABLKN	Meters	$\geq$30	30 Meters	30.0
SURFTN	°K	>-99.0	②	-99
PØTMPN	°K	>-99.0	②	-99
ZELEVN	Meters (MSL)	>-99.0	②	-99
DTLMDN	Fraction	>0 and <1	0.4	0.4
ISMTHN	Integer	$\geq$0 and $\leq$25	7	7
NCNTN	Integer	$\geq$0	2	2

(Table A-2 continued on following page.)

①If the variable is equal to or less than the initial value the variable is set equal to the default value.

②These variables are normally calculated from the EPAMS data base. They are given as input variables to provide the capability to override calculated quantities.

TABLE A-2 (Continued)

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE①	INITIAL VALUE
NETRDN	Index	≥ -2 and ≤ 4	2	-9
KFLG1	Switch	0 or 1	0	0
KFLG2	Switch	0 or 1	0 or if TYPE 3=9 then 1	0

①See previous page for footnote.

FORTRAN VARIABLE	DEFINITION
VN	Vector component v of the mean layer wind for use in the mesoscale wind field calculations in meters/second. The component v is positive toward the north and negative toward the south. (Default = calculated from the meteorological data base.) $VN = V \cdot \cos(\theta \pm 180.0)$
PN	Elevation (MSL) of the surface mixing layer above the meteorological station (if available) or source location in meters. (Default = calculated from the meteorological data base.)
GN	Reduced gravity parameter for use in the mesoscale wind field calculations in units of meters/second^2. See Page 6, Section 2 of the main body of this report. (Default = calculated from the meteorological data base.)
ABLKN	Minimum surface mixing layer depth constraint in meters. This parameter must be greater than zero. (Default = 30 meters.)
SURFTN	Surface temperature in degrees kelvin. (Default = determined from the meteorological data base.)
PØTMPN	Array containing the potential temperature at a maximum of 25 levels in degrees kelvin. (Default = determined from the meteorological data base.)
ZELEVN	Array containing the elevation (MSL) in meters of each of the values input into PØTMPN. (Default = determined from the meteorological data base.)
DTLMDN	Mesoscale model solution stability parameter shown in Equation 2-4 in the main body of this report. (Default = 0.4.)

FORTRAN VARIABLE	DEFINITION
ISMTHN	Number of mesoscale model time iterations between applications of a nine-point low-pass filter. (Default = 7, filtered every 7th step.)
NCNTN	Number of mesoscale model time iterations between recalculations of the model time increment constant. (Default = 2, recalculated every 2nd step.)
NETRDN	Net radiation index as defined in Section 3.2 in the main body of this report. (Default = calculated from the meteorological data base.)
KFLG1	Switch to control the initialization of the mesoscale model. If set equal to zero, the program will initialize the mesoscale model with a cold or hot start according to the rules given in Section 4.1 of the main body of this report. If KFLG1 is set equal to 1, the program forces a cold start, i.e., the model is initialized with the constant momentum field calculated from UN, VN and PN above. (Default = 0.)
KFLG2	Switch to control the print output from the Analysis Phase. If set equal to 0, only completion messages are output to the EPAMS output file XOTFIL (printer). If set non-zero, the Analysis Phase prints detailed tables of the meteorological data used and solution characteristics of the mesoscale model. If the absolute value of TYPE3 defined in Section A.1 is equal to 9, KFLG2 is automatically set non-zero. However, no mesoscale output is produced. (Default = 0.)

A.3 APPLICATION PHASE INPUTS

This section describes the variables that are input to the transport/diffusion section of the Application Phase. The variables given below are read only if TYPE3

from the executive is positive (greater than zero). These variables are read in the FORTRAN namelist format through namelist QLST2 via EPAMS input file XINFIL. A summary table of the input variables is given with their respective default values in Table A-3. Possible input variables are:

FORTRAN VARIABLE	DEFINITION
XSØUR	UTM easterly (x) coordinate of the source location in meters. (Default = value used in Analysis Phase or if continuous source 1 or 2 is specified = 356000 meters (ASARCO).)
YSØUR	UTM northerly (y) coordinate of the source location in meters. (Default = value used in Analysis Phase or if continuous source 1 or 2 is specified = 3520100 meters (ASARCO).)
ZSØUR	Elevation (MSL) of the surface at the source location in meters. (Default = value used in Analysis Phase or if continuous source 1 or 2 is specified = 1150.6 meters (ASARCO).)
Q	Source strength of source. Input in units of total material released if an instantaneous source or total material per second if a continuous source. (Default = 1 if an instantaneous source, = 2520 g/second SO_2 if continuous source 1 (ASARCO Copper), = 1764 g/second SO_2 if continuous source 2 (ASARCO lead).)
SIGXØ	Standard deviation of the alongwind source distribution after cloud stabilization for instantaneous sources only. Input in units of meters. (Default = 0.)
SIGYØ	Standard deviation of the crosswind source distribution after cloud stabilization for instantaneous sources only. Input in units of meters. (Default = 0.)

TABLE A-3

SUMMARY TABLE OF THE INPUT VARIABLES FOR THE APPLICATION PHASE INPUT THROUGH NAMELIST QLST2

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE②
XSØUR	Meters	$\geq$300000 and $\leq$450000	XPLACE
YSØUR	Meters	$\geq$3520000 and $\leq$3670000	YPLACE
ZSØUR	Meters	$\geq$1340	ZPLACE
Q	①	>0	=1 if KINDSØ = 0 =2520 g/sec. if KINDSØ = 1 =1764 g/sec. if KINDSØ = 2
SIGXØ	Meters	$\geq$0	0
SIGYØ	Meters	$\geq$0	0
SIGZØ	Meters	$\geq$0	0
H	Meters	$\geq$0	=0 if KINDSØ = 0 =252.4 if KINDSØ = 1 =186.5 if KINDSØ = 2

(Table A-3 continued on following page.)

①The source strength input unit is total material released for instantaneous sources (KINDSØ = 0) or total material released per second for continuous sources (KINDSØ > 0). If the source strength is input as grams for instantaneous sources or grams per second for continuous sources, the output concentration units are grams per cubic meter and dosage units are gram seconds per cubic meter. Dosage is calculated only for instantaneous sources.

②If the value of the variable is less than or equal to zero, the variable is set equal to the default value.

TABLE A-3 (Continued)

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE②
Z	Meters	≥ 0	0
TUA	Seconds	≥ 0	2.5
ALPHA	Fraction	>0	1
BETA	Fraction	>0	1
KINDSØ	Integer	≥ 0	=0 if TYPE3 $\geq$ 0 =1 if TYPE3 = -1 =2 if TYPE3 = -2
DECAY	Fraction	≥ 0	0
BLAMDA	Fraction	≥ 0	0
GAMMA	Fraction	≥ 0	1
STTEMP	^{o}K	$\geq$ Ambient Air	=366.3 if KINDSØ = 1 =352.4 if KINDSØ = 2
RADIUS	Meters	>0	=4.877 if KINDSØ = 1 =3.658 if KINDSØ = 2
EXTVEL	Meters/ Second	≥ 0	=2.1476 if KINDSØ = 1 =2.7512 if KINDSØ = 2
VS	Meters/ Second	≥ 0	0
CØNISS	Concentration Output Units	≥ 0	1×10^{-4}, 1×10^{-6} and 1×10^{-12}
DØSISS	Dosage Output Units	≥ 0	1×10^{-4}, 1×10^{-6} and 1×10^{-12}

(Table A-3 continued on following page.)

②See previous page for footnote.

TABLE A-3 (Continued)

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE②
TEMPSF	^oK	≥ 0	Historical Data
RTLAPS	^oK/Meter	≥ 0	Historical Data
KFLG1	Integer	=0 or 1	0
KFLG2	Integer	=0 or 1	0

② See previous page for footnote.

FORTRAN VARIABLE	DEFINITION
SIGZØ	Standard deviation of the vertical source distribution after cloud stabilization for instantaneous sources only. Input in units of meters. (Default = 0.)
H	Source emission release height above ground if a continuous source or effective height above ground after cloud stabilization for instantaneous sources. Input in meters. (Default = 0 if instantaneous, = 252.4 meters if continuous source 1 (ASARCO Copper), = 186.5 meters if continuous source 2 (ASARCO Lead).)
Z	Height in meters above ground at which concentration and/or dosage is to be calculated. (Default = 0.)
TAU	Source emission time in seconds for instantaneous sources only. (Default = 2.5 seconds.)
ALPHA	Crosswind diffusion coefficient. (Default = 1.)
BETA	Vertical diffusion coefficient. (Default = 1.)
KINDSØ	Source type switch, where if KINDSØ = 0, the program assumes an instantaneous source. If KINDSØ is greater than zero, the program assumes a continuous source. If set equal to 1, the program assumes a continuous source 1 which is the copper smelter stack at ASARCO, El Paso, Texas. If set equal to 2, the program assumes a continuous source which is the lead smelter stack at ASARCO El Paso, Texas. (Default = 0.)
DECAY	Coefficient of time dependent decay input as a fraction. (Default = 0.)
BLAMDA	Coefficient of precipitation scavenging input as a fraction. (Default = 0.)

FORTRAN VARIABLE	DEFINITION
GAMMA	Coefficient of vertical reflection input as the fraction of material reflected at the surface. (Default = 1.)
STTEMP	Stack effluent exit temperature in degrees Kelvin for continuous sources only. (Default = 366.3 °K if continuous source 1, = 352.4 if continuous source 2, no default is provided for KINDSØ > 2.)
RADIUS	Internal stack radius in meters for continuous sources only. (Default = 4.877 meters for continuous source 1, = 3.658 meters for continuous source 2. No default is provided for KINDSØ > 2.)
VS	Particle or droplet settling velocity in meters/second for concentration and/or dosage with depletion due to gravitational deposition. (Default = 0.)
CØNISS	Array of a maximum of 3 concentration values for which concentration isopleth half widths are to be calculated. (Default = 1×10^{-4}, 1×10^{-6} and 1×10^{-12}.)
DØSISS	Array of a maximum of 3 dosage values for which dosage isopleth half widths are calculated for instantaneous sources only. (Default = 1×10^{-4}, 1×10^{-6} and 1×10^{-12}.)
TEMPSF	Surface temperature in degrees Kelvin for continuous source only. (Default = calculated in Analysis Phase or from historical data.)
RTLAPS	Lapse rate of potential temperature in degrees Kelvin/meter for continuous sources only. (Default = calculated from elevation and potential temperature provided by the Analysis Phase or from historical data.)
KFLG1	Mesoscale model solution switch. If set equal to 0, the program will calculate the source cloud trajectory from the present or most current mesoscale

FORTRAN VARIABLE	DEFINITION
KFLG1 (Continued)	wind field solution. If non-zero, the program will use the past mesoscale solution. (Default = 0.)
KFLG2	Detail print output switch. If equal to zero, the program writes to the EPAMS output file the calculated concentration, dosage, standard deviation of the lateral concentration distribution and the isopleth half widths at each trajectory point. If non-zero, the program also outputs the trajectory points with the wind speed, surface layer depth, time and total distance. (Default = 0.)

A.4 GRAPHICS PHASE INPUTS

This section describes the variables that are input to the Graphics Phase. The variables given below are read only if TYPE3 from the executive is positive (greater than zero). These variables are read using the FORTRAN namelist format through namelist QLST3 via EPAMS file XINFIL. A summary table of the input variables is given with their respective default values in Table A-4. The plot routines assume all plot output is to be written to FORTRAN logical unit 1 on magnetic tape. Also, the plot routines assume the plotter is a drum plotter with at least 24 inch paper and with three different pen turrets to hold optional plot colors. The best color combination is for pen 1 to be black, pen 2 to be red and pen 3 to be blue.

FORTRAN VARIABLE	DEFINITION
ISW	Array of print and/or plot options
ISW(1)	If equal to zero, mesoscale wind field solutions are not printed. If equal to 1, mesoscale wind field solutions are printed in wind speed vector

TABLE A-4

SUMMARY TABLE OF THE INPUT VARIABLES FOR THE GRAPHICS PHASE INPUT THROUGH NAMELIST QLST3

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE ③
ISW(1)	N/A	$\geq$-2 and $\leq$2	0
ISW(2)	N/A	0 or 1	1
ISW(3)	N/A	0 or 1	0
ISW(5)	N/A	0 or 1	0
ISW(7)	N/A	0 or 1	0
ISW(8)	N/A	0 or 1	0
ISW(9)	N/A	$\geq$-2 and $\leq$2	0
ISW(10)	N/A	0, 1, 2 or -2	0
ISW(11)	N/A	0 or 1	0
XST	Meters	$\geq$300000 and $\leq$450000	XPLACE

(Table A-4 continued on following page.)

① Terrain contour intervals are defaulted to 1219.2, 1524, 1828.8, 2133.6, 2438.4, 2743.2, 3048, 3352.8 meters.

② Mixing layer depth contour intervals are defaulted to 50, 100, 200, 300, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600 meters.

③ If the value of the variable is less than or equal to zero the variable is set equal to its default value.

TABLE A-4 (Continued)

FORTRAN VARIABLE	UNITS	LIMITS	DEFAULT VALUE③
YST	Meters	$\geq$3520000 and $\leq$3670000	YPLACE
NCØNT	Integer	$\geq$0 and $\leq$8	8
CØNT	Meters	$\geq$0	①
NCØNTS	Integer	$\geq$0 and $\leq$16	16
CØNTS	Meters	$\geq$0	②
SCL	Inches/Meter	>0	1×10^{-4}
CNISS	Concentration Units	$\geq$0	N/A
DSISS	Dosage Units	$\geq$0	N/A

①Terrain contour intervals are defaulted to 1219.2, 1524, 1828.8, 2133.6, 2438.4, 2743.2, 3048, 3352.8 meters.

②Mixing layer depth contour intervals are defaulted to 50, 100, 200, 300, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600 meters.

③If the value of the variable is less than or equal to zero the variable is set equal to its default value.

FORTRAN VARIABLE	DEFINITION
ISW(1) (Continued)	components for each elapsed 30 minutes of mesoscale model time plus the final solution. If equal to 2, the wind speed and direction are printed rather than vector components. If negative, only the final wind field solution is printed. (Allow for 31 pages of output for each mesoscale wind field solution printed. The maximum is 155 pages.) (Default = 0.)
ISW(2)	If non-zero, isopleths of concentration and/or dosage are plotted on a terrain contour map. The data is taken from the transport/diffusion solution file. Also, isopleths can be recalculated and plotted by the use of CNISS and/or DSISS below. (Default = 1.)
ISW(3)	If non-zero, the relative variance of the change in momentum and the relative variance of the change in layer depth are plotted as a function of mesoscale model solution time. For an explanation of the purposes of this plot see Sections 4.2 and 4.3 of the main body of this report. (Default = 0.)
ISW(4)	Not used at present.
ISW(5)	If non-zero, the total momentum and total volume from the mesoscale solution are plotted as a function of mesoscale model solution time. For an explanation of the purpose of this plot, see Section 4.3 of the main body of this document. (Default = 0.)
ISW(6)	Not used at present.
ISW(7)	If non-zero, the program uses elevation (MSL) rather than depth in plotting contours of the mixing layer under ISW(10) below. (Default = 0.)
ISW(8)	If non-zero, a contour plot of the terrain elevation (MSL) is generated. (Default = 0.)

FORTRAN VARIABLE	DEFINITION
ISW(9)	If equal to 1, wind speed and direction vectors are plotted on a terrain contour map. If equal to 2, the wind speed and direction vectors are plotted without the terrain contours. If ISW(9) is positive, each intermediate mesoscale solution is plotted at 30 minute intervals including the final solution. A maximum of 5 plots could be generated under this option. However, if ISW(9) is negative, only the final solution is plotted. If this option is selected, specify 3 different pen colors for the three different pen turrets (black, red and blue). (Default = 0.)
ISW(10)	If equal to 1, contours of layer depth or elevation depending on ISW(7) are plotted on plots generated by ISW(9) above. If equal to 2, contours of layer depth or elevation are plotted on contours of terrain elevation for each of the intermediate mesoscale solutions. If equal to -2, only the final solution is plotted. (Default = 0.)
ISW(11)	If non-zero, trajectories generated from optional start locations given by XST and YST below are calculated and plotted on plots generated under ISW(9) above. If ISW(11) is positive, trajectories are plotted only. If negative, trajectories are both printed and plotted. (Default = 0.)
ISW(12) through ISW(19)	Not used at present.
XST	Array of a maximum of 5 UTM easterly (x) coordinates of the start location of optional trajectories in meters.
YST	Array of a maximum of 5 UTM northerly (y) coordinates of the start location of optional trajectories in meters.

FORTRAN VARIABLE	DEFINITION
NCØNT	Number of terrain elevation contours to generate. (Default = 8.)
CØNT	Array of a maximum of 8 contour elevation levels in meters. (Default = 1219.2 to 3352.8 in increments of 304.8 meters or 1000 feet.)
NCØNTS	Number of mixing-layer depth (elevation) contours to generate. (Default = 16.)
CØNTS	Array of a maximum of 16 layer depth (elevation) contour levels in meters. (Default = 50, 100, 200, 300, 400 to 2600 in increments of 200 meters.)
SCL	Plot scale factor for ISW(2), ISW(8), ISW(9), ISW(10) and ISW(11) in inches per meter. (Default = 1.0×10^{-4}.)
CNISS	Array of a maximum of 3 concentration isopleth values to use in place of those in the diffusion solution file under ISW(2) above. (No defaults provided.)
DSISS	Array of a maximum of 3 dosage isopleth values to use in place of those in the diffusion solution file under ISW(2) above. (No defaults provided.)

A.5 EXAMPLE EPAMS EXECUTION CARD DECK

Figure A-5 gives an example job card deck that executes the EPAMS program. The first three cards shown in Figure A-5 initialize the job and define the EPAMS program file. The next two cards are used to assign a plot output tape to be generated by the Graphics Phase of EPAMS. The sixth card shown assigns all of the catalogued data and solution files required by EPAMS and is an element residing in the EPAMS program file. The seventh card executes the data case

```
@FIN
▽$QLST3▽ISW=0,2*1,0,1,3*0,-1,2*1,▽$END
▽▽4▽▽▽▽▽1
▽$QLST2▽KFLG2=1,Q=100,▽$END
▽▽1▽▽1▽▽1
▽$QLST1▽KFLG2=1,▽$END
▽▽2▽▽1▽▽1304▽11▽13▽14▽00▽▽380000.▽▽3560000.0▽▽1310.0
@XQT▽B.ABSEPAMS
@ADD▽B.FILEPAMS
@MSG,W▽CØ10U,PAN=####,RUN=EPAMS,NAME=users name
@ASG,T▽1.,T,CØ101U
@USE▽B.,NEWHEC*EPAMS.
@ASG,AX▽NEWHEC*EPAMS.
@RUN▽EPAMS,PAN,name,time,pages
```

FIGURE A-5. Example UNIVAC 1108 EPAMS deck set up to calculate concentration and dosage from an instantaneous source on November 13, 1974 at 1400Z. The symbol ▽ means a blank punch.

cards. The first two data cards direct the Analysis Phase to analyse the meteorological data on November 13, 1974 at 1400Z. Also, the source location and elevation is given on data card one to specify the release point and enable the program to select the closest meteorological data. Optional input was selected for the Analysis Phase in order to set KFLG2 to 1 to obtain detailed output from this phase. The next two data cards direct EPAMS to execute, with detailed output, the Application Phase for the concentration and dosage from an instaneous source. The data cards give a source strength of 100 and allow the program to default all other possible inputs. The last two data cards execute the Graphics Phase and specify plots of concentration and dosage isopleths are to be produced, as well as plots of the final mesoscale wind field solution, relative variance of momentum and depth and total momentum and volume.

The order of the data cards in the example gives an EPAMS solution from start to end. However, the user could execute any one or group of the EPAMS phases depending on the solution desired. An example would be to place a set of Application and Graphics data cards after those shown in Figure 5-1 to produce concentration and dosage for an alternate source location. The user in this case would duplicate the last 4 data cards and would then include the alternate source location in the $QLST2 card of the second set.

A.6 DATA INPUT METHOD

The three phases of EPAMS use the FORTRAN namelist data input method. The namelist input data must be in a specific form in order to be read using a NAMELIST list. The first character in each card to be read must be blank. The first card in the namelist list contains the namelist name preceded by the character $ ($QLST1). The last card in each namelist list contains $END to terminate the list. The form of the remaining data items in the list may be:

a. *Variable Name = Constant* - The *variable name* may be a subscripted array name or a single variable name. Subscripts must be integer constants. The *constant* may be integer or real.

b. *Array Name = Set of Constants* (Separated by Commas) - The *array name* is not subscripted. The *set of constants* consists of constants of the type real or integer. The number of constants must be less than or equal to the array size. Successive occurrences of the same constant can be represent in the form k * *constant.*

The sequence of the input data parameters within the list is not significant. A more detailed explanation of the FORTRAN NAMELIST can be found in any Fortran Language Manual. The input parameters within the namelist that have default values are initialized to zero or other values in the Analysis Phase (Table A-2) prior to input of the first case. Parameters that are not used or have default constant or calculated values need not appear in the namelist list. When multiple cases are executed, all parameters retain their values from the previous case except in the Analysis Phase and are changed only by input. Also, the computer program assumes that core storage has been zeroed prior to execution.

A.7 EPAMS TEMPORARY EXECUTIVE AND DATA FILES

The H. E. Cramer Company has provided a temporary executive routine XECAMS to drive the programs that comprise the Analysis, Application and Graphics Phases of EPAMS. Program XECAMS is the main executive driver of EPAMS. This program assigns logical file numbers to each of the 12 EPAMS data and solution files. The internal file names are held in common block FILES defined on page B-8 of Appendix B and are assumed defined prior to the execution of any EPAMS phase. The internal file name, logical number and contents are

FILE NAME	LOGICAL UNIT NUMBER	CONTENTS
XINFIL	5	Card reader.
XØTFIL	6	Printer.
ØBFILE	0	Upper-air data master file directory. This file contains the station number, station location and hours for which upper air data has been recorded.
UAFILE	1	Upper-air data inventory file containing the observations for the stations and hours identified in ØBFILE.
SFCØBF	20	Surface data master file directory containing the station number, station location and hours for which surface data has been recorded.
SBFILE	21	Surface data inventory file containing the observation for the stations and hours identified in SFCØBF.
GWCPFL	10	GWC (Global Weather Central) data master file directory containing the location and rawinsonde (upper-air) hours for which GWC prediction data has been calculated.
GWCFIL	11	GWC data inventory file containing the prediction data for the 18 hours following each rawinsonde hour and for each location given in GWCPFL.
WSTRRN	12	White Sands Missile Range terrain elevation (MSL meters) data. This file contains the 41 x 41 (1681) point matrix of elevation data where the internal 31 x 31 points are real terrain data and the 5 outermost rows and columns are artificially filled with the minimum WSMR elevation. The data

FILE NAME	LOGICAL UNIT NUMBER	CONTENTS
		are stored starting with the most south-west corner and proceeding eastward, then moving one point north and repeating so the last point is the most north-east elevation. The real terrain data of the internal 31 x 31 points has a south-west UTM coordinate of 300000, 3520000 meters. The last of the real terrain to the east is at x = 450000 and to the north is y = 3670000 covering 150 kilometers in both directions. This file is treated as a read-only file by the EPAMS phases.
MXLYRF	13	This file contains the present and past solutions of the mesoscale wind-field model. The file contains two data blocks, both 5111 words long. Block 1 is the past solution and block 2 is the present or most current solution. Each block consists of the variables given in common block MXBUFR defined in the FORTRAN procedure listing in Appendix B, page B-9.
TRDFFL	14	This file contains the diffusion model solutions from the Application Phase. The file contains one block of 4090 words. The block contains the variables NCALC, IULIAN, IMØNTH, IDAY, IHØUR, NINUTE, XSØURC, YSØURC, ZSØURC, KINDS, CØNI (3), DØSI (3), DUMMY(664), XX (310), YY(310), SIGMY(310), CHIT (310), DØST(310), YCHIT(310, 3) and YDØST(310, 3). The variable NCALC is the total number of points in the cloud trajectory, IULIAN is the Julian day of the solution, IMONTH is the month, IDAY is the day, IHOUR is the hour, NINUTE is the minute and XSØURC, YSØURC and ZSØUR is the source location. Also,

FILE NAME	LOGICAL UNIT NUMBER	CONTENTS
		KINDS is the source type number, CØNI concentration isopleth levels, DØSI dosage isopleth levels and DUMMY is extra space for future use. XX and YY define the trajectory location, SIGMY is the standard deviation of the crosswind concentration (dosage) distribution at XX, YY, CHIT and DØST are the concentration and dosage and YCHIT and YDØST are the concentration and dosage isopleth halfwidths, respectively.
SPARE	9	This file is used to store intermediate output from the mesoscale wind-field model for subsequent printing and/or plotting by the Graphics Phase. The file can contain from 2 to 6 blocks of data. The last block is indicated by the first word in the block being negative. This block contains the relative variance of the change in momentum and depth for each model time step, as well as the total volume and momentum for each time step. Each block contains 5111 words where the first blocks are the variables in common MXBUFR defined on page B-9 of Appendix B. The last block is also the same as common MXBUFR except the common block SWELL defined in subroutine MYPRG (page B88, Appendix B) has replaced the arrays M, N and P in MXBUFR and the first word of the block is set to the negative of the total number of mesoscale model time steps. Also, the first blocks are written at approximately 30 minutes model time intervals until model convergence. The final solution is then written before the last data block of this file.

Included in the common block of internal file names are three variables that define the number of data blocks in the master directory files. These variables are MXBLKU, MXBLKG and MXBLKS for files ØBFILE, GWCPFL and SFCØBF, respectively. The Analysis Phase programs use these variables to start reading the master file directories with the most current data, where the most current data is assumed to be at the end of the files. In a real time environment, these variables would probably be set to 1 with the most current observations being the only data in the files. These variables are also initialized in the executive EXAMS on page B-22 of Appendix B.

APPENDIX B

Appendix B contains the complete FORTRAN listing of the EPAMS computer programs. Page B-2 shows the UNIVAC 1108 element (FILEPAMS) used to assign the necessary data files. Page B-3 and B-4 show the UNIVAC 1108 element (EPAMS-MAP/WSMR) used to collect the EPAMS programs into a segmented absolute program named ABSEPAMS. Pages B-5 through B-21 show the UNIVAC 1108 FORTRAN procedure used to define the common blocks used by the EPAMS programs. Also, the procedure defines all important variables used in the program. The remainder of the Appendix contains a FORTRAN listing of each program used in the EPAMS system. Also, the subroutines UADATA, XTRCUA, SATVP, SFCDTA, TESTFD, GWCDTA, BLKIN, BLKØUT and DRMRD are routines written and supplied by personnel at WSMR. However, some of these routines have been modified and are not interchangeable with previous copies of the routines.

Card images used to assign the EPAMS data files.

```
000001    000    @ASG,AX    8160*SWOBS.
000002    000    @ASG,AX    8160*SWUAD.
000003    000    @ASG,AX    8160*SWGWCP.
000004    000    @ASG,AX    8160*SWGWCD.
000005    000    @ASG,AX    8160*SWSFOB.
000006    000    @ASG,AX    8160*SWSFCD.
000007    000    @ASG,AX    HEC*WSTRRN.
000008    000    @ASG,AX    HEC*TRDFFL.
000009    000    @ASG,AX    HEC*MXLYRF.
000010    000    @ASG,AX    HEC*SPARE.
000011    000    @USE       F0.,8160*SWOBS.
000012    000    @USE       F1.,8160*SWUAD.
000013    000    @USE       F9.,HEC*SPARE.
000014    000    @USE       F10.,8160*SWGWCP.
000015    000    @USE       F11.,8160*SWGWCD.
000016    000    @USE       F12.,HEC*WSTRRN.
000017    000    @USE       F13.,HEC*MXLYRF.
000018    000    @USE       F14.,HEC*TRDFFL.
000019    000    @USE       F20.,8160*SWSFOB.
000020    000    @USE       F21.,8160*SWSFCD.
```

Card images used to collect the EPAMS programs into a segmented absolute program named ASBSEPAMS.

```
000001   014   @MAP,ISR ,B.MXOADTS
000002   014    DEF MXOADT
000003   014    REF BLKIN,IDIFTM,CLSSTN,NERR2$,NWDU$,NIO2$,NERR3$,SIN,COS,ALOG,XPRR
000004   014    REF NSTOP$,ALOG10,ATAN2,SQRT
000005   014    IN B.MXOADT,.OADATA,.XTRCOA,.SATVP
000006   014    END
000007   014   @MAP,ISR ,B.MXGWDTS
000008   014    DEF MXGWDT
000009   014    REF BLKIN,CLSSTN,IDIFTM,NWDU$,NIO2$,NIO1$,NERR3$,ALOG10,XPRR
000010   014    IN B.MXGWDT,.GWCDTA,.SATVP
000011   014    END
000012   014   @MAP,ISR ,B.MXSTRTS
000013   014    DEF MXSTRT
000014   017    REF BLKIN,BLKOUT,NWDU$,NIO2$,NERR2$,SQRT,NERR3$,ATAN2
000015   014    IN B.MXSTRT,.REC003
000016   014    END
000017   017   @MAP,IS ,B.ABSEPAMS
000018   017    NOT NR-A*RBLIB.PLOT/CAL835,NR-A*RBLIB.FACTOR/WSMR01
000019   017    NOT NR-A*RBLIB.NUMBER/CAL835,NR-A*RBLIB.SYMBOL/CAL835
000020   017    NOT NR-A*RBLIB.SCALE/CAL835,NR-A*RBLIB.AXIS/CAL835
000021   017    NOT NR-A*RBLIB.LINE/CAL835,NR-A*RBLIB.GRID
000022   017    LIB ASLABS*NPLOT.
000023   014    SEG A
000024   014    IN B.XECAMS,.BLKIN,.BLKOUT,.DRMRD
000025   014    IN MESAGE,TIMES,FILES
000026   014    SEG B*,(A)
000027   014    IN B.ANALIZ
000028   014    SEG C*,(B)
000029   014    IN B.MIXLYR,NEWPRM
000030   014    SEG D*,(C)
000031   014    IN B.MXOBAN,.CLSSTN,.IDIFTM,ABUFER,OBUFER,TDATA,WDATA,SFDATA,METPRM,GWDATA
000032   014    SEG E*,(D)
000033   014    IN B.MXOADTS
000034   014    SEG E1*,E
000035   014    IN B.MXGWDTS
000036   014    SEG E2*,E
000037   014    IN B.LYRAMM,.LYRDMM
000038   014    SEG E4*,E
```

```
014   IN B.MXSFDT,SBUFER
014   SEG F2*,(E4)
014   IN B.SFCDTA,.TESTFD
014   SEG F4*,F2
014   IN B.DAYNIT
014   SEG F6*,F2
014   IN B.LONLAT
014   SEG D1*,D
014   IN B.SOLVMX,MXBUFR,WNDFLD,WNDCOM
014   SEG E5*,(D1)
014   IN B.MXSTRTS
014   SEG E6*,E5
014   IN B.MXSOLV
014   SEG E7*,E5
014   IN B.MXCOUT
014   SEG F7*,(E6)
014   IN B.MYPRG,SWELL
014   SEG B3*,B
018   IN B.PLTPRG,.HEC014/WSMR,.HEC016/WSMR,.HEC015/WSMR,.HEC003,BS,M1,DTA1,N1
018   IN BUFRMX,CONDWN
014   SEG G*,(B3)
018   IN B.HEC004,.HEC013/WSMR
014   SEG G1*,G
014   IN B.HEC005,.HEC009
014   SEG G3*,G
014   IN B.HEC006,.HEC008
014   SEG G4*,G
014   IN B.HEC007
014   SEG B2*,B
015   IN B.APPLY
014   SEG C1*,(B2)
016   IN B.TRYDIF,DIFUZZ,SOURCE,COMMND
014   SEG D2*,(C1)
014   IN B.TRNPRT
014   SEG D3*,D2
014   IN B.DIFFUZ
014   END
017   @PACK B.
017   @PREP B.
017   @PRT,T B.
```

Card images for the UNIVAC 1108 Procedure Definition Processor (PDP) containing program common blocks.

```
      COMENT  PROC
      C*C*C*C*C*C XECAMS ANALYSIS, APPLICATION AND GRAPHICS PHASES C*C*C*C*C
      C
      C     THIS PROGRAM ASSUMES THE PROCS MESAGE, TIMES AND FILES GIVEN BELOW
      C     HAVE BEEN DEFINED IN THE CONTROL PROGRAMS THAT EXECUTE THESE
      C     ROUTINES. THE DEFINITIONS OF THE VARIABLES IN THESE PROCS ARE
      C     GIVEN BELOW. THE PROGRAM CONSISTS OF THREE PHASES ANALYSIS,
      C     APPLICATION AND GRAPHICS.
      C
      C    (NOTE - IF YOU ARE USING THE DUMMY COPY OF XECAMS THE INPUT DATA  )
      C    (        VARIABLES TYPE1,TYPE2,TYPE3,JULIAN,MONTH,DAY,HOUR,MINUTE, )
      C    (        XPLACE,YPLACE,ZPLACE  ARE INPUT FIRST ON EXECUTION AND    )
      C    (        THEN AFTER THE COMPLETION OF EACH PHASE OF PROCESSING. ALL)
      C    (        ARE INTEGERS EXCEPT THE LAST THREE. THEY ARE READ USING   )
      C    (        AN (8I3,3F10.2) FORMAT. THE VARIABLES ARE DEFINED IN PROC )
      C    (        MESAGE AND TIMES BELOW. THIS PROG IS INITIALLY PROVIDED   )
      C    (        WITH THE DUMMY ROUTINES XECAMS, ANALIZ, APPLY ALL OTHER   )
      C    (        ROUTINES ARE OPERATIONAL PROGRAMS.                        )
      C
      C     THE ANALYSIS PHASE IS EXECUTED BY A CALL TO SUBROUTINE MIXLYR.
      C     THIS PHASE ANALYZES THE METEROLOGICAL DATA BASE AND CALCULATES
      C     PARAMETERS NECESSARY FOR THE EXECUTION OF THE MESO-SCALE MODEL
      C     AND OTHER PARAMETERS FOR USE IN THE APPLICATION PHASE. THIS PHASE
      C     HAS OPTIONAL INPUT DATA IN NAMELIST FORMAT VIA NAMELIST QLST1.
      C     THIS IS DETERMINED BY THE VARIABLE TYPE3 DEFINED IN PROC MESAGE
      C     BELOW. THIS PHASE STORES ALL RESULTS IN THE DATA BASE AND AT
      C     COMPLETION RETURNS CONTROL TO THE CALLING PROGRAM. IF THE PROG
      C      CANNOT COMPLETE THE MIXING LAYER SOLUTION OR THE MESOSCALE
      C     SOLUTION TYPE3 IS RETURNED AS A NEGATIVE NUMBER.
      C     ONLY THE MIXING LAYER CALCULATIONS PART OF THE ANALYSIS SECTION
      C     IS EXECUTED IF TYPE3 IS SET EQUAL TO 9. SEE PROC MESAGE BELOW.
      C
      C     THE APPLICATION PHASE IS EXECUTED BY A CALL TO SUBROUTINE TRYDIF.
      C     THIS PHASE OBTAINS THE RESULTS OF THE ANALYSIS PHASE FROM THE DATA
      C     BASE AND EXECUTES THE TRANSPORT (TRAJECTORY) AND DIFFUSION MODELS.
      C     THIS PHASE HAS OPTIONAL INPUT DATA IN NAMELIST FORMAT VIA
      C     NAMELIST QLST2. THIS IS DETERMINED BY THE VARIABLE TYPE3 DEFINED
      C     IN PROC MESAGE BELOW. THIS PHASE PRINTS THE DIFFUSION MODEL
```

```
C     CALCULATIONS AND STORES ALL RESULTS IN THE DATA BASE AND RETURNS
C     CONTROL TO THE CALLING PROGRAM. IF THE APPLICATION PHASE DOES NOT
C     HAVE ENOUGH INFORMATION OR IF THERE IS A MODEL FAILURE TYPE3 IS
C     RETURNED AS A NEGATIVE NUMBER. ALSO, TYPE3 CAN BE USED TO
C     DEFAULT A SOURCE TYPE.
C
C     THE GRAPHICS PHASE IS EXECUTED BY A CALL TO SUBROUTINE PLTPRG.
C     THIS PHASE OBTAINS THE RESULTS OF THE ANALYSIS OR THE APPLICATION
C     PHASE OR BOTH AND PRINTS AND/OR PLOTS THE RESULTS. THIS PHASE
C     HAS OPTIONAL INPUT DATA IN NAMELIST FORMAT VIA NAMELIST GLST3.
C     THIS IS DETERMINED BY THE VARIABLE TYPE3 DEFINED IN PROC MESAGE
C     BELOW. AT THE COMPLETION OF THIS PHASE CONTROL IS RETURNED TO THE
C     CALLING PROGRAM. IF THE APPLICATION PHASE CANNOT COMPLETE AN
C     ASSIGNED TASK TYPE3 IS RETURNED AS A NEGATIVE NUMBER.
C **  FILE INFORMATION - FILES USED IN DIFFERENT PHASES
C     NOTE - FILE LOGICAL NUMBER ASSIGNMENTS ARE MADE IN XECAMS
C     APPLICATION PROG USES FILES MXLYRF,WSTRRN, TROFFL, XINFIL, XOTFIL
C    -FILE MXLYRF CONTAINS 2 BLOCKS 5111 WORDS PER BLOCK, MESOSCALE
C     SOLUTION FILE CONTAINING PAST (BLOCK 1) AND PRESENT (BLOCK 2)
C     SOLUTIONS. EACH BLOCK IS CONSTRUCTED AS IN COMMON MXBUFER
C    -FILE WSTRRN  CONTAINS THE TERRAIN DATA (41X41 GRID) (1 BLOCK)
C    -FILE TROFFL  CONTAINS THE DIFFUSION MODEL SOLUTION (1 BLOCK OF
C     4090 WORDS. THIS BLOCK CONTAINS NCALC,IULIAN,IMONTH,IDAY,IHOUR,
C     MINUTE,XSOURC,YSOURC,ZSOURC,KINDS,
C     CONI(3),DOSI(3),DUMMY(664),XX(310),YY(310),SIGMY(310),CHIT(310),
C     DOST(310),YCHIT(310,3),YDOST(310,3)  NCALC IS THE NO. OF TRAJECT
C     POINTS - IULIAN TO MINUTE COME FROM PROC TIMES -
C     XSOURC,YSOURC,ZSOURC IS THE SOURCE LOCATION - KINDS IS THE
C     SOURCE TYPE (KINDSO BELOW) - CONI,DOSI INPUT ISOPLETH VALUES
C     (CONISS,DOSISS) - XX,YY TRAJECTORY COORDINATES - SIGMY, CHIT,
C     DOST ARE SIGMA-Y, CON AND DOS AT XX,YY - YCHIT AND DOST ARE THE
C     LATERAL DISTANCE FROM CENTERLINE TO ISOPLETH VALUE FOR THE 3 INPUT
C     ISOPLETH LEVELS.
C
C     ANALYSIS - PROG USES FILES MXLYRF, WSTRRN, OBFILE, UAFILE
C     GWCFIL,GWCPFL,SBFILE,SFCOBF, SPARE, XINFIL, XOTFIL
C    -FILE SPARE CONTAINS UP TO 5 BLOCKS OF INTERMEDIATE MESOSCALE
C     SOLUTIONS FROM COMMON MXBUFER. THE LAST BLOCK CONTAINS 1000 WORDS
```

```
C      CONTAINING INFO TO PLOT MOMENTUM, VOLUME, REL CHANGE MOM AND REL
C      CHANGE VOLUME (SPARE IS WRITTEN IN SUB MYPRG)
C
C      GRAPHICS - PROG USES FILES MXLYRF, WSTRRN, TREFFL, SPARE, XINFIL,
C                 XOTFIL
C
       END
MESAGE   PROC
       COMMON /MESAGE/ TYPE1,TYPE2,TYPE3,XPLACE,YPLACE,ZPLACE
       INTEGER TYPE1,TYPE2,TYPE3
C **   THESE VARIABLES ARE PASSED TO THE PROCESSING PHASES. VIA THIS
C      COMMON BLOCK. THE PHASES ASSUME -
C      TYPE1 - NOT REFERENCED
C      TYPE2 - NOT REFERENCED
C      TYPE3 - INPUT DATA CONTROL VARIABLE REFERENCED IN ALL THREE PHASES
C         ANALYSIS PHASE - IF TYPE3 IS ZERO NO INPUT DATA IS ASSUMED.
C                          IF TYPE3 IS NON-ZERO INPUT DATA IS READ FROM
C                          UNIT XINFIL VIA NAMELIST QLST1. THE VARIABLES
C                          THAT CAN BE INPUT ARE DEFINED IN PROC NEWPRM
C                          BELOW. TYPE3 IS RETURNED AS A NEG NO. IF
C                          THERE IS A FAILURE IN THE ANALYSIS PHASE.
C                          IF THE ABSOLUTE VALUE OF TYPE3 = 9 THE MESO-
C                          SCALE SOLUTION IS NOT PRODUCED, ONLY THE
C                          MIXING LAYER SOLUTION IS PRODUCED. THIS
C                          OPTION IS USED ONLY TO DISPLAY THE MET DATA
C                          USED AND MIXING LAYER CALCULATIONS. IF TYPE3
C                          IS NEG. NO INPUT DATA IS ASSUMED. IF POS.
C                          INPUT DATA IS READ AS ABOVE. HOWEVER, KFLG2
C                          IN NAMELIST QLST1 (PROC NEWPRM) IS NOW SET
C                          NON-ZERO AUTOMATICALLY.
C         APPLICATION PHASE - IF TYPE3 IS ZERO OR NEGATIVE NO INPUT DATA
C                             IS ASSUMED AND ALL INPUT PARAMETERS ARE
C                             DEFAULTED. HOWEVER, IF TYPE3 = -1 THE PROG
C                             DEFAULTS TO CONTINUOUS SOURCE 1 (ASARCO
C                             COPPER), OR IF TYPE3 = -2 THE PROGRAM
C                             DEFAULTS TO CONTINUOUS SOURCE 2 (ASARCO
C                             LEAD).
C                             IF TYPE3 IS > ZERO INPUT DATA IS READ
```

```
C                                       FROM UNIT XINFIL VIA NAMELIST QLST2. THE
C                                       VARIABLES THAT CAN BE INPUT ARE DEFINED
C                                       IN PROC SOURCE BELOW. TYPE3 IS RETURNED AS
C                                       A NEG IF THERE IS A FAILURE IN THE
C                                       APPLICATION PHASE.
C            GRAPHICS PHASE - IF TYPE3 IS A ZERO NO INPUT DATA IS ASSUMED.
C                             IF TYPE3 IS NON-ZERO INPUT DATA IS READ
C                             FROM UNIT XINFIL VIA NAMELIST QLST3. THE
C                             VARIABLES THAT CAN BE INPUT ARE DEFINED
C                             IN PROC N1 BELOW. TYPE3 IS RETURNED AS A NEG
C                             IF THERE IS A FAILURE IN THE GRAPHICS PHASE.
C        XPLACE - X (EASTERLY) UTM COORDINATE OF THE SOURCE (EMISSION)
C                 LOCATION IN METERS (IF = 0.0 THE PROG USES 360000.0
C                 FROM PROC DEFINP BELOW) (REFERENCED ONLY IN THE ANALYSIS
C                 PHASE)
C        YPLACE - Y (NORTHERLY) UTM COORDINATE OF THE SOURCE (EMISSION)
C                 LOCATION IN METERS (IF = 0.0 THE PROG USES 3585000.0
C                 FROM PROC DEFINP BELOW) (REFERENCED ONLY IN THE ANALYSIS
C                 PHASE)
C        ZPLACE - Z ELEVATION OF SOURCE (EMISSION) BASE LOCATION IN METERS
C                 (IF = 0.0 THE PROGRAM USES 1310.0 FROM PROC DEFINP BELOW)
C                 (REFERENCED ONLY IN THE ANALYSIS PHASE)
 END
FILES   PROC
        COMMON /FILES/ XINFIL,XOTFIL,SPARE,WSTRRN,OBFILE,MXLYRF,TRDFFL
      1,UAFILE,GWCFIL,GWCPFL,SBFILE,SFCOBF,MXBLKU,MXBLKG,MXBLKS
        INTEGER XINFIL,XOTFIL,SPARE,WSTRRN,OBFILE,MXLYRF,TRDFFL
        INTEGER UAFILE,GWCFIL,GWCPFL,SBFILE,SFCOBF
C **    THESE VARIABLES ARE PASSED TO THE 3 PHASES VIA THIS COMMON BLOCK
C       XINFIL - INPUT DATA (CARD) FILE
C       XOTFIL - OUTPUT (PRINT) FILE
C       SPARE - FILE USED TO STORE TEMPORY OUTPUT FROM ANALYSIS PHASE FOR
C                 PRINTING AND/OR PLOTTING BY GRAPHIC PHASE
C       WSTRRN - TERRAIN DATA FILE
C       OBFILE - UPPER AIR DATA MASTER DIRECTORY FILE
C       MXLYRF - MESO-SCALE SOLUTION FILE
C       TRDFFL - DIFFUSION MODEL SOLUTION FILE
C       UAFILE - UPPER AIR DATA INVENTORY FILE
```

```
C      GWCFIL - GWC DATA INVENTORY FILE
C      GWCPFL - GWC DATA MASTER DIRECTORY FILE
C      SBFILE - SURFACE DATA INVENTORY FILE
C      SFCOBF - SURFACE DATA MASTER DIRECTORY FILE
C      MXBLKO - MAXIMUM NO. OF DATA BLOCKS IN FILE OBFILE
C      MXBLKG - MAXIMUM NO. OF DATA LOCKS IN FILE GWCPFL
C      MXBLKS - MAXIMUM NO. OF DATA BLOCKS IN FILE SFCOBF
 END
MXBUFR  PROC
       COMMON /MXBUFR/ NETRAD,IULIAN,IMONTH,IDAY,IHOUR,NINUTE,XSOURC,
      1YSOURC,ZSOURC,SURFTA,POTMPR(25),ZELEVR(25),UO,VO,PO,ABLKO,DTLMDO,
      2GO,ISMTHO,NCNTO,M(41,41),N(41,41),P(41,41)
       REAL M,N
C      MXBUFR CONTAINS THE WINDFIELD SOLUTION MATRICES M,N,P AND THE
C      INITIALIZATION AND CONTROL PARAMETERS USED IN THE SOLUTION
C      NETRAD - NET RADIATION INDEX
C      XSOURC - UTM EASTERLY COORDINATE (METERS) OF SOURCE
C      YSOURC - UTM NORTHERLY COORDINATE (METERS) OF SOURCE
C      ZSOURC - ELEVATION OF SOURCE (METERS)
C      SURFTA - CONTAINS THE SURFACE TEMP
C      POTMPR - CONTAINS THE POTENTIAL TEMP AT THE FIRST 25 LEVELS
C      ZELEVR - CONTAINS THE ELEVATION OF THE FIRST 25 LEVELS
C      UO - U COMPONENT OF WIND SPEED (MPS)
C      VO - V COMPONENT OF WIND SPEED (MPS)
C      PO - LAYER ELEVATION (METERS)
C      ABLKO - MINIMUM LAYER DEPTH (METERS) CONSTRAINT
C      DTLMDO - STABILITY PARAMETER FOR CALCULATING TIME STEP BETWEEN
C               MODEL ITERATIONS
C      GO - REDUCED GRAVITY FACTOR
C      ISMTHO - NO. OF TIME STEPS (ITERATIONS) BETWEEN FILTER APPLICATION
C      NCNTO - NO. OF TIME STEPS (ITERATIONS) BETWEEN RECALCULATION
C              OF THE TIME STEP INCREMENT
 END
BUFRMX  PROC
C      THIS COMMON BLOCK IS THE SAME AS MXBUFR ABOVE IT IS RENAMED FOR
C      USE IN SUBROUTINE PLTPRG FOR EASE OF SEGMENTED COLLECTION
       COMMON /BUFRMX/ NETRAD,IULIAN,IMONTH,IDAY,IHOUR,NINUTE,XSOURC,
      1YSOURC,ZSOURC,SURFTA,POTMPR(25),ZELEVR(25),UO,VO,PO,ABLKO,DTLMDO,
```

```
         2G0,ISMTHO,NCNTO,M(41,41),N(41,41),P(41,41)
          REAL M,N
      END
      WNDFLD  PROC
          COMMON /WNDFLD/ IJM,JDM,G2,DTL,NN,ISM,TIM,ICNT,UVMIN,UVEGU,UVMAX,
         1DPMAX,GP(39),B3P(39),C3P(39),DELXI(6),DELYJ(6),DXPI(6),DYPJ(6),
         2TIMMAX,IFLG,STPN,H(41,41)
C
C
C **  NOTE - ALL VARIABLES WITH ** DEPEND ON THE TERRAIN GRID MATRIX
C **          OR TERRAIN GRID AXES. IF EITHER ARE MODIFIED REVIEW ALL
C **          ** CARDS FOR NECESSARY MODIFICATIONS. ALSO, REVIEW COMMENTS
C **          IN SUBROUTINE TRNPRT.
C
C
C     IJM,JDM,G2,DTL,NN,ISM,TIM,ICNT - INTERNAL CONSTANTS AND COUNTERS
C     UVMIN - MIN. VALUE > 0 FOR U OR V TO PREVENT UNDERFLOWS
C     UVMAX - MAX. VALUE FOR U OR V TO PREVENT MODEL INSTABILITY
C     UVEGU - USED IN CALCULATING M AND N AND P TO PREVENT UNDERFLOWS
C     DPMAX - MAX. DEPTH OF LAYER TO PREVENT MODEL INSTABILITY
C     GP,B3P,C3P, - ARRAYS USED TO STORE FORWARD DIFFERENCES IN MODEL
C                   CALCULATION TO AVOID CALCULATING BACKWARD DIFFERENCES
C **  DELXI - ARRAY CONTAINING THE INVERSE OF THE POS. DISTANCE BETWEEN
C              POINTS 1-3,2-4,3-5,4-6,5-7,6-8 OF THE EXPANDED GRID OF THE
C              X AXIS. THE ARRAY DELXI IS SET THROUGH DATA IN PROC WNDDTA
C              BELOW AND EQUALS (1/(48*XINC)),(1/(24*XINC)),(1/(12*XINC))
C              ,(1/(6*XINC)),(1/(3*XINC)),(1/(2*XINC))
C **  DELYJ - ARRAY CONTAINING THE INVERSE OF THE POS. DISTANCE BETWEEN
C              POINTS 1-3,2-4,3-5,4-6,5-7,6-8,OF THE EXPANDED GRID OF THE
C              Y AXIS. THE ARRAY DELYJ IS SET THROUGH DATA IN PROC WNDDTA
C              BELOW IN THE SAME MANNER AS DELXI EXCEPT YINC IS USED.
C **  DXPI - ARRAY CONTAINING THE INVERSE OF THE POS. DISTANCE BETWEEN
C              EACH POINT 1-2,2-3,3-4,4-5,5-6,6-7 OF THE EXPANDED GRID OF
C              THE X AXIS. THE ARRAY DXPI IS SET THROUGH DATA IN PROC
C              WNDDTA BELOW AND EQUALS (1/(32*XINC)),(1/(16*XINC)),
C              (1/(8*XINC)),(1/(4*XINC)),(1/(2*XINC)),(1/XINC)
C **  DYPJ - SAME AS DXPI EXCEPT FOR Y AXIS AND USES YINC.
C
```

```
C      TIMMAX - MAX MODEL TIME FOR ANY RUN.
C      IFLG,STPN - CONTROL PARAMETERS
C **   H - ELEVATION OF TERRAIN - THIS ARRAY IS INPUT AS 1 BLOCK FROM
C            FILE WSTRRN AND CONSISTS OF THE REGULAR EVENLY SPACED GRID
C            PLUS THE EXPANDED GRID REGION. THE EXPANDED GRID REGION
C            (1ST AND LAST 5 ROWS AND COLUMNS) IS SET EQUAL TO THE MINIMUM
C            TERRAIN ELEVATION (HMIN IN PROC WNDCOM) UNLESS SEVERE TERRAIN
C            GRADIENTS ARE CAUSING STABILITY PROBLEMS. IN PLACES OF SEVERE
C            TERRAIN GRADIENTS THE FIFTH ROW (COLUMN) IN SHOULD BE SET TO
C            ITS REAL TERRAIN VALUE IN THESE ISOLATED AREAS.
C         PROG CURRENTLY READS 1681 (41,41) POINTS INTO H
  END
WNDCOM  PROC
       COMMON /WNDCOM/ IDP,JDP,XSTRT,YSTRT,XINC,YINC,XMIN,HMIN
C **   IDP - THE NUMBER OF TOTAL PNTS. IN THE X AXIS OR THE NO. OF PNTS.
C              OF REAL TERRAIN PLUS 10 FOR THE EXPANDED REGION AT BOTH ENDS
C              OF THE AXIS. (CURRENTLY SET TO 41)
C **   JDP - THE NUMBER OF TOTAL PNTS. IN THE Y AXIS OR THE NO. OF PNTS.
C              OF REAL TERRAIN PLUS 10 FOR THE EXPANDED REGION AT BOTH ENDS
C              OF THE AXIS. (CURRENTLY SET TO 41)
C **   XSTRT - STARTING COORDINATE OF THE X AXIS OF THE REAL TERRAIN GRID
C              (NOT INCLUDING EXPANDED REGION) (METERS)
C **   YSTRT - STARTING COORDINATE OF THE Y AXIS OF THE REAL TERRAIN GRID
C              (NOT INCLUDING EXPANDED REGION) (METERS)
C **   XINC,YINC - INCREMENT IN METERS BETWEEN GRID POINTS IN THE REAL
C              TERRAIN GRID AREA FOR THE X,Y AXES  (METERS)
C **   XMIN - MINIMUM VALUE OF XINC AND YINC (METERS)
C **   HMIN - MINIMUM TERRAIN ELEVATION HEIGHT (METERS)
  END
COMWND  PROC
C      THIS COMMON BLOCK IS THE SAME AS WNDCOM ABOVE IT IS RENAMED FOR
C      USE IN SUBROUTINE TRYDIF FOR EASE OF SEGMENTED COLLECTION
       COMMON /COMWND/ IDP,JDP,XSTRT,YSTRT,XINC,YINC,XMIN,HMIN
  END
CONDWN  PROC
C      THIS COMMON BLOCK IS THE SAME AS WNDCOM ABOVE IT IS RENAMED FOR
C      USE IN SUBROUTINE PLTPRG FOR EASE OF SEGMENTED COLLECTION
       COMMON /CONDWN/ IDP,JDP,XSTRT,YSTRT,XINC,YINC,XMIN,HMIN
```

```
      END
WNDDTA  PROC
C **  VALUES OF GRID PARAMETERS THAT ARE CONSTANT OVER THE GRID REGION
       DATA DELXI/4.1666666E-6,8.3333333E-6,1.6666666E-5,3.3333333E-5,
     16.6666666E-5,1.0E-4/
       DATA DELYJ/4.1666666E-6,8.3333333E-6,1.6666666E-5,3.3333333E-5,
     16.6666666E-5,1.0E-4/
       DATA DXPI/6.25E-6,1.25E-5,2.5E-5,5.0E-5,1.0E-4,2.0E-4/
       DATA DYPJ/6.25E-6,1.25E-5,2.5E-5,5.0E-5,1.0E-4,2.0E-4/
       DATA TIMMAX/7200./,IFLG/0/
      END
WNDDTB  PROC
C **  VALUES OF PROGRAM CONSTANTS THAT ARE FIXED BY THE TERRAIN GRID
C      USED
       DATA IDP,JDP,XSTRT,YSTRT,XINC,YINC,XMIN,HMIN/41,41,300000.,
     13520000.,3*5000.,1140./
      END
TIMES   PROC
C **  THESE VARIABLES ARE PASTED TO THE ANALYSIS PHASE VIA THIS COMMON
C      BLOCK. THESE VARIABLES GIVE THE TIME AND DATE FOR THE DESIRED
C      SOLUTION. THE TIME AND DATE CAN BE PAST, PRESENT OR FUTURE
C      JULIAN - JULIAN DAY OF LAST DAY OF PREVIOUS MONTH
C      MONTH - MONTH (1-12)
C      DAY - DAY (1-31)
C      HOUR - HOUR IN GREENWICH TIME (00-23)
C      MINUTE - MINUTE (00-60)
       COMMON /TIMES/ JULIAN,MONTH,DAY,HOUR,MINUTE
       INTEGER DAY,HOUR
      END
N1      PROC
C **  THIS PROC DEFINES DATA INPUT TO THE GRAPHICS PHASE. DATA IS READ
C      FROM UNIT XINFIL VIA NAMELIST GLST3 ONLY IF TYPE3 IN PROC MESAGE
C      IS NON-ZERO. IF NO INPUT IS DESIRED THE PROGRAM DEFAULTS THE
C      VARIABLES AS GIVEN BELOW.
C         ISW - ARRAY OF PRINT/PLOT OUTPUT OPTIONS
C      ISW(1) - IF = 0 MESO-SCALE WIND FIELD SOLUTIONS ARE NOT PRINTED,
C               IF = 1 PRINT IN VECTOR COMPONENTS. IF = 2 PRINT IN SPEED
C               AND DIRECTION. IF POSITIVE ALL INTERMEDIATE SOLUTIONS
```

```
C                ARE PRINTED (30,60,90,120, FINAL MINUTES). IF NEGATIVE
C                ONLY THE FINAL SOLUTION IS PRINTED. ( DEFAULT = 0)
C     ISW(2) - IF≠ 0 PLOT THE DIFFUSION ISOPLETHS OR RECALCULATE
C                ISOPLETHS AND PLOT ON TERRAIN MAP (DEFAULT = 1)
C     ISW(3) - IF ≠ 0 PLOT THE RELATIVE VARIANCE OF THE CHANGE IN
C                MOMENTUM AND VOLUME FROM THE MESO-SCALE SOLUTION TO
C                ANALYZE BEHAVIOR OF SOLUTION. (DEFAULT = 0)
C     ISW(4) - N/A
C     ISW(5) - IF ≠ 0 PLOT TOTAL MOMENTUM AND VOLUME VERSUS TIME FROM
C                THE MESO-SCALE SOLUTION TO ANALYZE THE BEHAVIOR OF THE
C                SOLUTION (DEFAULT = 0)
C     ISW(6) - N/A
C     ISW(7) - IF ≠ 0 THE PROGRAM USES ELEVATION RATHER THAN DEPTH IN
C                ISW(10) BELOW. IF ≠ 0 YOU MUST INPUT NCONTS AND CONTS
C                BELOW (DEFAULT = 0)
C     ISW(8) - IF ≠ 0 PLOT TERRAIN ELEVATION MAP FROM FILE WSTRRN
C                (USES ONLY INTERNAL 31X31 GRID)  (DEFAULT = 0)
C     ISW(9) - IF = 1 PLOT VECTOR MESO-SCALE SOLUTIONS ON TERRAIN MAP,
C                IF = 2 PLOT VECTOR MESO-SCALE SOLUTIONS ALONE,
C                IF POSITIVE ALL INTERMEDIATE SOLUTIONS ARE PLOTTED (30,60,
C                90,120, FINAL MINUTES) IF NEGATIVE ONLY THE FINAL
C                SOLUTION IS PLOTTED. (DEFAULT =0)
C     ISW(10) - IF = 1 PLOT THE LAYER DEPTH ON PLOTS GENERATED BY ISW(9)
C                 IF = 2 PLOT THE LAYER DEPTH ON TERRAIN ONLY. IF EQUAL TO
C                 2 AND POSITIVE ALL INTERMEDIATE SOLUTIONS ARE PLOTTED,
C                 IF = 2 AND NEGATIVE ONLY THE FINAL SOLUTION IS PLOTTED.
C                 (DEFAULT = 0)
C     ISW(11) - IF ≠ 0 CALCULATE TRAJECTORIES FROM START POINTS XST,YST
C                 AND PLOT ON PLOTS GENERATED BY ISW(9) ABOVE. IF POSITIVE
C                 PLOT TRAJECTORIES ONLY. IF NEGATIVE PRINT AND PLOT
C                 TRAJECTORIES (DEFAULT = 0)
C     ISW(12)-ISW(19) - N/A
C     ISW(20) - DO NOT USE, INTERNAL PROG SWITCH
C     XST,YST - ARRAYS OF START POINTS OF TRAJECTORIES GENERATED UNDER
C                 ISW(11) ABOVE. (MAX OF 5 VALUES EACH). IF XST AND YST
C                 ARE BOTH ZERO THE PROG DEFAULTS TO THE SOURCE LOCATION
C                 USED IN THE ANALYSIS PHASE.
C     NCONT - NO. OR TERRAIN CONTOUR LEVELS (DEFAULT = 8)
```

```
C      CONT = ARRAY OF TERRAIN CONTURS (DEFAULT = 1219.2 TO 3352.8 IN
C              INCREMENTS OF 304.8 M (1000 FEET))
C      NCONTS - NO. OF LAYER DEPTH (ELEVATION) CONTOUR LEVELS (DEFAULT=16)
C      CONTS - ARRAY OF LAYER DEPTH (ELEVATION) CONTOUR LEVELS
C              (DEFAULT=50,100,200,300,400 TO 2600 IN INC. OF 200 M)
C      SCL - PLOT SCALE FACTOR IN INCHES/METER (DEFAULT = 1.0E-4)
C      CNISS - ARRAY OF OPTIONAL CONCENTRATION ISOPLETH LEVELS
C              (MAX OF 3 VALUES, IF INPUT THEN PROG. DOES NOT PLOT THOSE
C              ISOPLETH VALUES CALCULATED IN DIFFUSION PHASE)
C      DSISS - ARRAY OF OPTIONAL DOSAGE ISOPLETH LEVELS
C              (MAX OF 3 VALUES, IF INPUT THEN PROG. DOES NOT PLOT THOSE
C               ISOPLTH VALUES CALCULATED IN DIFFUSION PHASE)
C
       COMMON /N1/ ISW(20),XST(5),YST(5),CONT(8),NCONT,CONTS(16),NCONTS,
      1IST,IND,JST,JND,RUNID(10),IEVEN,NCRV,SCL,IDIM,JDIM
 END
SOURCE  PROC
C **   THIS PROC DEFINES DATA INPUT TO THE APPLICATION PHASE. DATA IS
C      READ FROM UNIT XINFIL VIA NAMELIST QLST2 ONLY IF TYPE3 IN PROC
C      MESAGE ABOVE IS NON-ZERO. IF NO INPUT IS DESIRED OR VARIABLES ARE
C      NOT IN INPUT LIST THEY DEFAULT TO VALUES GIVEN BELOW.
C      XSOUR - UTM EASTERLY COORDINATE OF RELEASE POINT (SOURCE) (METERS)
C              (DEFAULT = THAT USED IN ANALYSIS IF INST. SOURCE, OR
C                = 356000.0 IF CONT. SOURCE 1 OR 2)
C      YSOUR - UTM NORTHERLY COORDINATE OF RELEASE PNT (SOURCE) (METERS)
C              (DEFAULT = THAT USED IN ANALYSIS IF INST. SOURCE , OR
C                = 3520100.0 IF CONT. SOURCE 1 OR 2)
C      ZSOUR - ELEVATION AT GROUND OF SOURCE (METERS) (DEFAULT = THAT
C              USED IN ANALYSIS IF AN INST. SOURCE, OR 1150.6 M IF CONT.
C              SOURCE 1, OR 2)
C      Q - SOURCE STRENGTH OF SOURCE - INPUT IN UNITS OF TOTAL MATERIAL
C          RELEASED IF AN INSTANTANEOUS SOURCE - INPUT IN UNITS OF TOTAL
C          MATERIAL PER SECOND IF A CONTINUOUS SOURCE.
C                    CONCENTRATION - UNITS/CUBIC METER
C                    DOSAGE - UNITS SECONDS/CUBIC METER
C          (DEFAULT = 1 IF AN INST. SOURCE, = 2520 G/SEC SO2 IF CONT.
C           SOURCE 1, = 1764 G/SEC SO2 IF CONT. SOURCE 2)
C      SIGXO - STANDARD DEVIATION OF THE ALONGWIND SOURCE DISTRIBUTION
```

```
C              AFTER CLOUD STABILIZATION (METERS)
C              INSTANTANEOUS SOURCE ONLY (DEFAULT = 0.0)
C     SIGYO - STANDARD DEVIATION OF THE CROSSWIND SOURCE DISTRIBUTION
C              AFTER CLOUD STABILIZATION (METERS)
C              INSTANTANEOUS SOURCE ONLY (DEFAULT = 0.0)
C     SIGZO - STANDARD DEVIATION OF THE VERTICAL SOURCE DISTRIBUTION
C              AFTER CLOUD STABILIZATION (METERS)
C              INSTANTANEOUS SOURCE ONLY (DEFAULT = 0.0)
C     H - HEIGHT ABOVE GROUND (ZSOUR) OF CLOUD AFTER STABILIZATION
C          FOR INSTANTANEOUS SOURCES, OR HEIGHT ABOVE GROUND OF
C          EMISSION FOR CONTINUOUS SOURCES (METERS)
C          (DEFAULT = 0 FOR INST. SOURCE, = 252.4 M IF CONT. SOURCE 1,
C           = 186.5 M IF CONT. SOURCE 2)
C     Z - HEIGHT ABOVE GROUND FOR CONCENTRATION AND DOSAGE CALCULATIONS
C          (METERS) (DEFAULT = 0)
C     TAU - SOURCE EMISSION TIME FOR INSTANTANEOUS SOURCES (SECONDS)
C             (DEFAULT = 2.5 SEC)
C     ALPHA - CROSSWIND DIFFUSION COEFICIENT (DEFAULT = 1)
C     BETA - VERTICAL DIFFUSION COEFFICIENT (DEFAULT = 1)
C     KINDSO - IF = 0 INSTANTANEOUS SOURCE, IF ≠ 0 CONTINUOUS SOURCE.
C               IF = 1 CONTINUOUS SOURCE 1 (ASARCO COPPER)
C               IF = 2 CONTINUOUS SOURCE 2 (ASARCO LEAD)
C               (IF TYPE3 >= 0 DEFAULT = 0) (IF TYPE3 = -1 OR -2 DEFAULT
C               = 1 OR 2)
C     DECAY - COEFFICIENT OF TIME DEPENDENT DECAY (DEFAULT = 0)
C     BLAMDA - COEFFICIENT OF PRECIPITATION SCAVENGING (DEFAULT = 0)
C     GAMMA - REFLECTION COEFFICIENT - FRACTION OF MATERIAL REFLECTED
C              AT THE SURFACE (DEFAULT = 1.0)
C     STTEMP - STACK EFFLUENT EXIT TEMPERATURE (DEG K) CONTINUOUS SOURCE
C               ONLY
C               (DEFAULT = 366.3 DEG K CONT. SOURCE 1, OR = 352.4 DEG K
C               CONT. SOURCE 2)
C     RADIUS - INTERNAL STACK RADIUS (METERS) CONTINUOUS SOURCES ONLY
C               (DEFAULT = 4.877 M CONT. SOURCE 1, OR = 3.658 M CONT.
C               SOURCE 2)
C     EXTVEL - STACK EFFLUENT EXIT VELOCITY (METERS/SECOND) CONTINUOUS
C               SOURCES ONLY
C               (DEFAULT = 2.1476 M/SEC CONT. SOURCE 1, OR 2.7512 M/SEC
```

```
C                 CONT. SOURCE 2)
C         VS - PARTICLE SETTLING VELOCITY FOR GRAVITATIONAL DEPOSITION
C              OR DEPLETION OF CONCENTRATION AND DOSAGE DUE TO GRAVITATIONAL
C              DEPOSITION (METERS/SEC)
C              (DEFAULT = 0)
C         CONISS - ARRAY OF 3 VALUES CONTAINING THE CONCENTRATION ISOPLETH
C                  LEVELS IN UNITS OF UNITS PER CUBIC METER (DEFAULT =
C                  1.0E-4, 1.0E-6, 1.0E-12)
C         DOSISS - ARRAY OF 3 VALUES CONTAINING THE DOSAGE ISOPLETH
C                  LEVELS IN UNITS OF UNITS SECONDS PER CUBIC METER (DEFAULT
C               = 1.0E-4, 1.0E-6, 1.0E-12)
C         TEMPSF - SURFACE TEMP (DEG K) IF NOT PROVIDED BY ANALYSIS PHASE
C                  (CONTINUOUS SOURCES ONLY)
C         RTLAPS - LAPSE RATE OF POTENTIAL TEMPERATURE (DEG K/METER) IF NOT
C                  PROVIDED BY THE ANALYSIS PHASE (CONTINUOUS SOURCES ONLY)
C         KFLG1 - IF = 0 THE PROGRAM WILL USE THE PRESENT SOLUTION OF THE
C                 MESO-SCALE MODEL IN THE APPLICATION PHASE. IF ≠ 0 THE
C                 PROGRAM WILL USE THE PAST MESO-SCALE SOLUTION.
C         KFLG2 - IF = 0 ONLY CONCENTRATION, DOSAGE AND ISOPLETHS ARE
C                 PRINTED. IF ≠ 0 THE TRAJECTORY AND RELATED PARAMETERS ARE
C                  ALSO PRINTED.
C
          COMMON /SOURCE/ XSOUR,YSOUR,ZSOUR,Q,SIGXO,SIGYO,SIGZO,H,Z,TAU
         1,ALPHA,BETA,KINDSO,DECAY,BLAMDA,GAMMA,STTEMP,RADIUS,EXTVEL,VS,
         2CONISS(3),DOSISS(3),TEMPSF,RTLAPS,KFLG1,KFLG2
  END
DIFUZZ  PROC
C         PARAMETERS PAST TO DIFFUSION MODEL ROUTINE
C         NETRAD IS THE CALCULATED NET RADIATION INDEX FROM ANALYSIS ROUTINE
C         XSOURC IS THE RELEASE (SOURCE) LOCATION USED BY ANALYSIS ROUTINES
C         YSOURC IS THE RELEASE (SOURCE) LOCATION USED BY ANALYSIS ROUTINEW
C         ZSOURC IS THE ELEVATION OF THE RELEASE (SOURCE) USED BY ANALYSIS
C                ROUTINES (ELEVATION OF GROUND)
C         XX - ARRAY OF UTM X COORDINATES IN METERS DEFINING THE TRAJECTORY
C              OF THE CLOUD CENTROID
C         YY - ARRAY OF UTM Y COORDINATES IN METERS DEFINING THE TRAJECTORY
C              OF THE CLOUD CENTROID
C         UBAR - ARRAY GIVING THE LAYER WIND SPEED IN MPS AT EACH XX,YY PNT.
```

```
C      HM - ARRAY GIVING THE MIXING LAYER DEPTH IN METERS AT EACH XX,YY
C            POINT
C      R - SCRATCH SPACE USED IN CONCENTRATION, ETC CALCULATIONS
C      SURFTA - SURFACE TEMP FROM ANALYSIS PHASE (DEG K)
C      POTMPR - POTENTIAL TEMP AT EACH LEVEL ZELEVR (DEG K)
C      ZELVEVR - ELEVATION (METERS)
C
       COMMON /DIFUZZ/ NETRAD,IULIAN,IMONTH,IDAY,IHOUR,NINUTE,XSOURC,
      1YSOURC,ZSOURC,SURFTA,POTMPR(25),ZELEVR(25),XX(310),YY(310),
      2UBAR(310),HM(310),R(3611)
C
C **   THE COMMON AREA DIFUZZ MUST BE GREATER THAN OR EQUAL IN SIZE TO
C      COMMON MXBUFR ABOVE. ALSO, WITH A GRID SPACING OF 5 KM THE PROG
C      USES 1/10 OR 500 M AS THE TRAJECTORY INCREMENT GIVING A MAXIMUM
C      OF 310 POINTS FROM ONE GRID SIDE TO THE OTHER. IF THE GRID SPACE
C      IS REDUCED FROM 5 KM TO A SMALLER VALUE THE ARRAY IN DIFFUZ
C      MAY HAVE TO BE INCREASED. IF SO SEE ALSO, REQUIRED CHANGES IN
C      SUBROUTINE TRNPRT
 END
NEWPRM  PROC
C
C ** THIS PROC DEFINES DATA INPUT TO THE ANALYSIS PHASE. DATA IS
C      READ FROM UNIT XINFIL VIA NAMELIST QLST1 ONLY IF TYPE3 IN PROC
C      MESAGE ABOVE IS NON-ZERO. IF NO INPUT IS DESIRED OR VARIABLES ARE
C      NOT IN INPUT LIST THEY DEFAULT TO VALUES GIVEN BELOW.
C
C      UN - U COMPONENT OF THE LAYER WIND SPEED (MPS) (UN IS CALCULATED
C            IF NOT INPUT)
C      VN - V COMPONENT OF THE LAYER WIND SPEED (MPS) (VN IS CALCULATED
C            IF NOT INPUT)
C      PN - ELEVATION OF THE MIXING LAYER ABOVE THE STATION OR SOURCE
C            (METERS) (PN IS CALCULATED IF NOT INPUT)
C      GN - REDUCED GRAVITY PARAMETER FOR MESO-SCALE MODEL (DEFAULT =
C            CALCULATED)
C      ABLKN - MINIMUM MIXING LAYER DEPTH CONSTRAINT (DEFAULT = 30 M)
C      SURFTN - SURFACE TEMPERATURE (DEGREES K) (DEFAULT = MET DATA BASE)
C      POTMPN - ARRAY CONTAINING THE POTENTIAL TEMP (DEG K) AT THE FIRST
C                 25 LEVELS (DEFAULT = MET DATA BASE)
```

```
C      ZELEVN - ARRAY CONTAINING THE ELEVATION (M) OF THE FIRST 25 LEVELS
C                 (DEFAULT = MET DATA BASE)
C      DTLMDN - MESO-SCALE MODEL STABILITY PARAMETER (DEFAULT = 0.4)
C      ISMTHN - NO. OF MESO-SCALE MODEL ITERATIONS BETWEEN APPLICATION
C                 OF THE 9 POINT FILTER (DEFAULT = 7)
C      NCNTN - NO. OF MESO-SCALE MODEL ITERATIONS BETWEEN RECALCULATIONS
C                 OF THE ITERATION TIME INTERVAL (DEFAULT = 2)
C      NETRDN - NET RADIATION INDEX (DEFAULT = MET DATA BASE)
C      KFLG1 - IF = 0 THE PROGRAM WILL USE A COLD OR A HOT START IN
C               THE MESO-SCALE MODEL DEPENDING ON WHETHER UN,VN, PN ARE
C               WITHIN 25% OF VALUES USED IN PREVIOUS SOLUTION.
C               IF ≠ 0 THE PROGRAM WILL FORCE A COLD START OF MESO-SCALE
C               MODEL.
C      KFLG2 - IF = 0 THE PROGRAM WILL PRINT ONLY COMPLETION MESSAGES
C               IF ≠ 0 THE PROGRAM WILL PRINT DETAILED DATA MESSAGES
       COMMON /NEWPRM/ UN,VN,PN,GN,ABLKN,DTLMDN,ISMTHN,NCNTN,NETRDN,
      1XSOURN,YSOURN,ZSOURN,SURFTN,POTMPN(25),ZELEVN(25),IFLG1,KFLG1,
      2KFLG2
   END
OBUFER  PROC
C      UPPER AIR DATA INVENTORY
C      UASTNS - STATION NUMBER
C      UASTNA
C      UASTNX - STATION UTM X COORD IN KM
C      UASTNY - STATION UTM Y COORD IN KM
C      UASTNZ - STATION ELEVATION IN METERS
C      TLVLMX - NO. OF LEVELS OF TEMP. DATA FOR STATION,RUN
C      TADRES - DRUM ADDRESS OF TEMP DATA FOR STATION,RUN
C      WLVLMX - NO. OF LEVELS OF WIND DATA FOR STATION ,RUN
C      WADRES - DRUM ADDRESS OF WIND DATA FOR STATION ,RUN
C      LATEST - RUN INDEX FOR LATEST RUN
C      RUNMAX - MAX. NO. OF RUNS
C      STNMAX - MAX. NO. OF STATIONS
       COMMON /OBUFER/ UASTNS(25),UASTNA(25),UASTNX(25),UASTNY(25),
      1UASTNZ(25),TLVLMX(25,3),TADRES(25,3),WLVLMX(25,3),WADRES(25,3)
      2,UATIME(3),LATEST,RUNMAX,STNMAX
       INTEGER TLVLMX,WLVLMX,TADRES,WADRES,LATEST,RUNMAX,STNMAX,UATIME,
      1UASTNS
```

```
      REAL UASTNX,UASTNY,UASTNZ,UASTNA
  END
TDATA   PROC
C     UPPER AIR DATA FOR PARTICULAR STATION, RUN
C     PS - PRESSURE IN PASCALS (PASCALS*.01 = MB)
C     QS -
C     TS - TEMP DEG K
C     ZS - ELEVATION METERS
C     TV - VIRTUAL TEMP DEG K
C     TA - POTENTIAL TEMP. DEG. K
C     US,VS - WIND COMPONENTS MPS
C     DELTA - HYDROSTATIC EQUATION EXPONENT
C     PD - D-VALUE FOR PRESSURE LEVEL
C     IMAX - NO. OF LEVELS
      COMMON /TDATA/ PS(40),QS(40),TS(40),ZS(40),TV(40),TA(40),US(40),
     1VS(40),DELTA(40),PD(40),IMAX
      REAL PS,PD,QS,TS,ZS,TV,TA,US,VS,DELTA
      INTEGER IMAX
  END
WDATA  PROC
C     UPPER AIR DATA -CONT-
C     UW,VW - WIND COMPONENTS MPS
C     ZW - ELEVATION METERS
C     KMAX - NO. OF LEVELS
      COMMON /WDATA/ UW(40),VW(40),ZW(40),KMAX
  END
ABUFER  PROC
C     SCRATCH AREA
      COMMON /ABUFER/ A(5700)
  END
GWDATA  PROC
C     GWC PREDICTION DATA
C     GTIME - TIME
C     P - PRESSURE (PASCALS)
C     T - TEMP. DEG. K
C     O - SPECIFIC HUMIDITY
C     U,V - WIND COMPONENTS (MPS)
C     Z - ELEVATION METERS
```

```
          COMMON /GWDATA/ GTIME,P(4),Q(4),T(4),U(4),V(4),Z(4)
          INTEGER GTIME
   END
SFDATA  PROC
C         SURFACE DATA
C         STIME - TIME
C         WD - WIND DIRECTION (DEG)
C         WS - SPEED (MPS)
C         WG - WIND GUSTS (MPS)
C         SLP - SEA LEVEL PRESS (MB) (CONVERTED TO STATION PRESS. IN MB)
C         BT - BAROMETRIC TENDENCY
C         TSS - SURFACE TEMP (DEG K)
C         TD - DEW POINT DEPRESSION (DEG K)
C         AS - ALTIMETER SETTING (INCHES HG)
C         PA - 6 HOURLY PRECIP. (MILLIMETERS)
C         TSC - TOTAL SKY COVER
C         PW - PAST WEATHER (WMO 4500)
C         VV - VISIBILITY (M)
C         WW1,WW2,WW3,WW4 - PRESENT WEATHER
C         NH - FRACTION OF CELESTIAL DOME COVERED BY LOW CLOUDS (8 THS)
C         CL - TYPE OF LOW CLOUD (WMO 0513)
C         H - HEIGHT ABOVE GROUND OF LOWEST CLOUD (WMO 1600)
C         CM - TYPE OF MIDDLE CLOUD (WMO 0515)
C         CH - TYPE OF HIGH CLOUD (WMO 0509)
C         NS - AMOUNT OF CLOUDS (8 THS)
C         K - CLASSIFICATION OF CLOUDS ACCORDING TO HEIGHT
C         CT - TYPE OF CLOUD WITHIN HEIGHT CLASSIFICATION
C         HSHS - HEIGHT ABOVE GROUND OF BASE OF CLOUD LAYER (WMO 1677)
          COMMON /SFDATA/ STIME,WD,WS,WG,SLP,BT,TSS,TD,AS,PA,TSC,PW,VV,WW1,
         1WW2,WW3,WW4,NH,CL,H,CM,CH,NS,K,CT,HSHS
          INTEGER STIME,TSC,PW,VV,WW1,WW2,WW3,WW4,CL,H,CM,CH,CT,HSHS,BT
   END
METPRM  PROC
          COMMON /METPRM/ IFLAG1,IFLAG2,IFLAG3,IFLAG4,I1,J1,K1,IPASS,NUMHRS,
         1NSTAIN,IDXSTA(25),STADST(25),STATNZ,STATNX,STATNY,TIME,NLVLT,T850,
         2SFSZ,SAVSPD,SAVDIR,OBUFRN,UABLK,GWCBLK,SFCBLK,MXUBLK,MXGBLK,MXSBLK
         3,JUTIME,N850,N700
   END
```

```
DEFINP  PROC
C      DEFAULT VALUES FOR INPUTS IN COMMON NEWPRM
       DATA UN,VN,PN,GN,ABLKN,DTLMDN,ISMTHN,NCNTN,NETRDN,XSOURN,YSOURN,
      1ZSOURN,SURFTN,POTMPN,ZELEVN,IFLG1,KFLG1,KFLG2/4*-99.0,30.0,
      20.4,7,2,-9,360000.0,3585000.0,1310.0,51*-99.0,3*0/
 END
GBUFER PROC
       COMMON /GBUFER/ GRIDX(25),GRIDY(25),GRDBLK(80),ANLTIM(80),RUNMAX,
      1GRDMAX
       INTEGER GRDBLK,ANLTIM,RUNMAX,GRDMAX
 END
SBUFER PROC
       COMMON /SBUFER/ SFCSTN(76),SFSTNA(76),SFSTNX(76),SFSTNY(76),
      1SFSTNZ(76),SADRES(76,6),RUNTIM(6),LATEST,RUNMAX,STNMAX
       INTEGER SFCSTN,SADRES,RUNTIM,LATEST,RUNMAX,STNMAX
 END
TRPDTA PROC
       COMMON /TRPDTA/ PT(3),QT(3),TT(3),ZT(3),TB(3),TI(3),UT(3),VT(3),
      1TD(3),NMAX
 END
```

```
C     MAIN PROGRAM XECAMS
C
      INCLUDE MESAGE                                                    XEC00100
      INCLUDE TIMES                                                     XEC00200
      INCLUDE FILES                                                     XEC00300
C     SET UP FILES AND INITIALIZE ALL APPROPRIATE CONSTANTS             XEC00400
      XINFIL = 5                                                        XEC00500
      XOTFIL = 6                                                        XEC00600
      OBFILE = 0                                                        XEC00700
      UAFILE = 1                                                        XEC00800
      SFCOBF = 20                                                       XEC00900
      SBFILE = 21                                                       XEC01000
      GWCPFL = 10                                                       XEC01100
      GWCFIL = 11                                                       XEC01200
      WSTRRN = 12                                                       XEC01300
      MXLYRF = 13                                                       XEC01400
      TROFFL = 14                                                       XEC01500
      SPARE = 9                                                         XEC01600
      MXBLKU = 18                                                       XEC01700
      MXBLKG = 1                                                        XEC01800
      MXBLKS = 120                                                      XEC01900
    5 READ (XINFIL,1000,END=90) TYPE1,TYPE2,TYPE3,JULIAN,MONTH,DAY,HOUR,XEC02000
     1MINUTE,XPLACE,YPLACE,ZPLACE                                       XEC02100
 1000 FORMAT (8I3,3F10.2)                                               XEC02200
   10 GO TO (20,60,40,80),TYPE1                                         XEC02300
C     CALL APPLICATION SUPERVISOR                                       XEC02400
   20 CALL APPLY                                                        XEC02500
      IF (TYPE3 .LT. 0) GO TO 90                                        XEC02600
      GO TO 5                                                           XEC02700
C     CALL ALTERNATE PROGS                                              XEC02800
   40 CONTINUE                                                          XEC02900
C     CALL OPYQC                                                        XEC03000
      GO TO 5                                                           XEC03100
C     CALL ANALYSIS SUPERVISOR                                          XEC03200
   60 CALL ANALIZ                                                       XEC03300
      IF (TYPE3 .LT. 0) GO TO 90                                        XEC03400
      GO TO 5                                                           XEC03500
C     WHAT NEXT                                                         XEC03600
```

```
   80 CONTINUE                                                   XEC03700
      CALL PLTPRG                                                XEC03800
      IF (TYPE3 .LT. 0) GO TO 90                                 XEC03900
      GO TO 5                                                    XEC04000
   90 STOP                                                       XEC04100
      END                                                        XEC04200
```

```
      SUBROUTINE ANALIZ                                               ANA00100
C     ANALYSIS SUPERVISOR                                             ANA00200
      INCLUDE MESAGE                                                  ANA00300
      GO TO (20,20,30),TYPE2                                          ANA00400
C     CALL MIXING LAYER ANALYSIS FOREMAN                              ANA00500
   20 CALL MIXLYR                                                     ANA00600
      GO TO 40                                                        ANA00700
C     OTHER ANALYSIS ROUTINES                                         ANA00800
   30 CONTINUE                                                        ANA00900
   40 RETURN                                                          ANA01000
      END                                                             ANA01100
```

```
      SUBROUTINE APPLY                                                  APY00100
C     THIS SUBROUTINE SUPERVISES THE APPLICATIONS ROUTINES              APY00200
      INCLUDE MESAGE,LIST                                               APY00300
C     SET UP APPROPRIATE CONSTANTS, ETC                                 APY00400
      GO TO (20,40,40),TYPE2                                            APY00500
C     CALL THE TRANSPORT/DIFFUSION APPLICATION FOREMAN                  APY00600
   20 CALL TRYDIF                                                       APY00700
      GO TO 60                                                          APY00800
C     OTHER APPLICATIONS FOREMAN                                        APY00900
   40 CONTINUE                                                          APY01000
   60 RETURN                                                            APY01100
      END                                                               APY01200
```

```
      SUBROUTINE MIXLYR                                                 MIX00100
C     THIS SUBROUTINE IS THE MIXING LAYER ANALYSIS FOREMAN              MIX00200
      INCLUDE MESAGE,LIST                                               MIX00300
      INCLUDE NEWPRM,LIST                                               MIX00400
C     CALCULATE WIND SPEED COMPONENTS AND LAYER HEIGHT                  MIX00500
  100 CALL MXOBAN                                                       MIX00600
      IF (TYPE3 .LT. 0) RETURN                                          MIX00700
      IF (TYPE3 .EQ. 9) RETURN                                          MIX00800
C     CALCULATE 2-LAYER MESOSCALE MODEL SOLUTION                        MIX00900
  200 CALL SOLVMX                                                       MIX01000
      RETURN                                                            MIX01100
      END                                                               MIX01200
```

```
      SUBROUTINE MXOBAN                                                 MX000100
      INCLUDE TIMES,LIST                                                MX000200
      INCLUDE ABUFER,LIST                                               MX000300
      INCLUDE FILES,LIST                                                MX000400
      INCLUDE NEWPRM,LIST                                               MX000500
      INCLUDE OBUFER,LIST                                               MX000600
      INCLUDE TDATA,LIST                                                MX000700
      INCLUDE WDATA,LIST                                                MX000800
      INCLUDE SFDATA,LIST                                               MX000900
      INCLUDE METPRM,LIST                                               MX001000
      INCLUDE DEFINP,LIST                                               MX001100
      INCLUDE MESAGE,LIST                                               MX001200
      DIMENSION KULIAN(12)                                              MX001300
      INTEGER TIME,UABLK,GWCBLK,SFCBLK,OBUFRN                           MX001400
      DATA MXUBLK,MXGBLK,MXSBLK/30,1,120/                               MX001500
      DATA KULIAN/0,31,59,90,120,151,181,212,243,273,304,334/           MX001600
C                                                                       MX001700
      NAMELIST /QLST1/ UN,VN,PN,GN,ABLKN,SURFTN,POTMPN,ZELEVN,DTLMDN,   MX001800
     1ISMTHN,NCNTN,NETRDN,KFLG1,KFLG2                                   MX001900
C                                                                       MX002000
C                                                                       MX002100
      IF (UN .LE. -99.0) GO TO 6                                        MX002200
      UN = -99.0                                                        MX002300
      VN = -99.0                                                        MX002400
      PN = -99.0                                                        MX002500
      GN = -99.0                                                        MX002600
      ABLKN = 30.0                                                      MX002700
      DTLMDN = 0.4                                                      MX002800
      ISMTHN = 7                                                        MX002900
      NCNTN = 2                                                         MX003000
      NETRDN = -9                                                       MX003100
      XSOURN = 360000.0                                                 MX003200
      YSOURN = 3585000.0                                                MX003300
      ZSOURN = 1310.0                                                   MX003400
      SURFTN = -99.0                                                    MX003500
      IFLG1 = 0                                                         MX003600
      KFLG1 = 0                                                         MX003700
      KFLG2 = 0                                                         MX003800
```

```
      DO 5 I=1,25                                                          MX003900
      POTMPN(I) = -99.0                                                    MX004000
    5 ZELEVN(I) = -99.0                                                    MX004100
    6 CONTINUE                                                             MX004200
      IF (TYPE3 .LE. 0) GO TO 10                                           MX004300
      READ (XINFIL,QLST1,END=10)                                           MX004400
   10 TYPE3 = IABS(TYPE3)                                                  MX004500
      IPASS = 74                                                           MX004600
      TIME = 0                                                             MX004700
      IF (MONTH .GT. 0.AND.DAY .GT. 0) GO TO 30                            MX004800
      CALL ERTRAN(9,UATIME(1),UATIME(2))                                   MX004900
      DECODE(12,9007,UATIME,L) MONTH,DAY,IPASS,HOUR,MINUTE,TIME            MX005000
      JULIAN = KULIAN(MONTH)                                               MX005100
      IF (MONTH.GT.2.AND.MOD(IPASS,4).EQ.0) JULIAN = JULIAN+1              MX005200
   30 UATIME(1) = JULIAN+DAY                                               MX005300
      UATIME(2) = HOUR*100+MINUTE                                          MX005400
      WRITE (XOTFIL,9006) IPASS,UATIME(1),MONTH,DAY,UATIME(2)              MX005500
      IF (TYPE3 .EQ. 9) KFLG2 = 1                                          MX005600
      IF (KFLG2 .GT. 0) WRITE (XOTFIL,QLST1)                               MX005700
      IF (MXBLKU .GT. 0) MXUBLK = MXBLKU                                   MX005800
      IF (MXBLKG .GT. 0) MXGBLK = MXBLKG                                   MX005900
      IF (MXBLKS .GT. 0) MXSBLK = MXBLKS                                   MX006000
      IF (XPLACE .LE. 0.0) XPLACE = XSOURN                                 MX006100
      IF (YPLACE .LE. 0.0) YPLACE = YSOURN                                 MX006200
      IF (ZPLACE .LE. 0.0) ZPLACE = ZSOURN                                 MX006300
      XSOURN = XPLACE                                                      MX006400
      YSOURN = YPLACE                                                      MX006500
      ZSOURN = ZPLACE                                                      MX006600
      SFSZ = ZSOURN                                                        MX006700
      SAVSPD = -99.0                                                       MX006800
      SAVDIR = -99.0                                                       MX006900
      UBRLYR = -99.0                                                       MX007000
      VBRLYR = -99.0                                                       MX007100
C                                                                          MX007200
C                                                                          MX007300
C     THIS SUBROUTINE CONTROLS THE CALCULATION OF THE MIXING LAYER DEPTH   MX007400
C     , THE LAYER WIND SPEED COMPONENTS AND THE NET RADIATION INDEX.       MX007500
C                                                                          MX007600
```

```
C     CALC TIME OF SOLUTION DESIRED                                      MX007700
      TIME = (DAY+JULIAN)*100+HOUR                                       MX007800
C     DETERMINE STATION AND DATA FOR RAWINSONDE DATA                     MX007900
      OBUFRN = 431                                                       MX008000
C      LOOK FOR DATA FOR PAST 3 HOURS                                    MX008100
      IPASS = 1                                                          MX008200
      CALL MXUADT                                                        MX008300
      IF (IFLAG1 .GT. 0) GO TO 100                                       MX008400
C     3 HOUR NOT FOUND TRY FOR 12 HOURS                                  MX008500
      IPASS = 2                                                          MX008600
      CALL MXUADT                                                        MX008700
C     IFLAG1 = 1 IS UPPER AIR DATA <= 3 HOURS OLD                        MX008800
C     IFLAG1 = 2 IS UPPER AIR DATA <= 12 HOURS OLD                       MX008900
C     IFLAG1 = 0 MEANS NO UPPER AIR DATA (DEFAULT HMLYR - LAYER DEPTH)   MX009000
      IF (IFLAG1 .EQ. 0) SFSZ = ZSOURN                                   MX009100
  100 CONTINUE                                                           MX009200
      IFLAG2 = 0                                                         MX009300
      IF (IFLAG1 .EQ. 1.OR.IFLAG1 .EQ. 0) GO TO 200                      MX009400
C     DETERMINE IF THERE IS GWC DATA                                     MX009500
      OBUFRN = 212                                                       MX009600
C     IFLAG2 = 0 NO GWC DATA                                             MX009700
C     IFLAG2 =  1 GWC DATA PRESENT NO MOD OF RAWIN AT 700 MB, ETC.       MX009800
C     IFLAG2 =  2 OR 3 GWC DATA PRESENT MOD RAWIN AT 700 MB AND PEPLACE  MX009900
C              RAWIN WINDS AT 850 AND 700 MB                             MX010000
C              IF = 3 THEN ALSO, ΔT 700 MB > 0                           MX010100
      CALL MXGWDT(PS,TS)                                                 MX010200
  200 CONTINUE                                                           MX010300
C     DETERMINE SURFACE DATA                                             MX010400
      OBUFRN = 845                                                       MX010500
      CALL MXSFDT                                                        MX010600
C     IFLAG3 = 0 NO SURFACE DATA, DEFAULT NET RADIATION INDEX            MX010700
C     IFLAG3 = 1 SURFACE DATA                                            MX010800
C     IFLAG3 = 2 RAWIN MODIFIED WITH SURFACE DATA                        MX010900
C     IFLAG3 = 3 SAME AS 2 BUT ALSO, MODIFIED WITH GWC AT 850 MB         MX011000
C     IFLAG3 = 4 SAME AS 3 BUT ALSO, ΔT AT 850 MB > 0                    MX011100
      IF (SURFTN .LE. 0.0.AND.TS(1) .GT. 0.0) SURFTN = TS(1)             MX011200
      IF (IFLAG1 .EQ. 0) GO TO 500                                       MX011300
      IF (IDIFTM(TIME,JUTIME) .LE. 6) GO TO 480                          MX011400
```

```
      IF (IFLAG3 .LT. 3) IFLAG1 = 0                                    MX011500
  480 CONTINUE                                                          MX011600
C     CALCULATE MIXING LAYER DEPTH                                      MX011700
      CALL LYRAHM(UBRLYR,VBRLYR,HMLYR)                                  MX011800
      IF (HMLYR .LE. 0.0.OR.HMLYR .GE. 9999.0) GO TO 500                MX011900
      IF (PN .LE. -99.0) PN = HMLYR                                     MX012000
      GO TO 530                                                         MX012100
C     DEFAULT MIXING LAYER HEIGHT - HMLYR                               MX012200
  500 CALL LYRDHM(HMLYR,SFSZ)                                           MX012300
      IF (PN .LE. -99.0) PN = HMLYR                                     MX012400
      IFLAG4 = 1                                                        MX012500
      GO TO 600                                                         MX012600
  530 CONTINUE                                                          MX012700
C     CALC GN                                                           MX012800
      IF (GN .GT. -0.1) GO TO 600                                       MX012900
      SUM1 = 0.0                                                        MX013000
      SUM2 = 0.0                                                        MX013100
      L = 0                                                             MX013200
  540 L = L+1                                                           MX013300
      IF (L .GT. NLVLT) GO TO 590                                       MX013400
      TLAST = TS(L)                                                     MX013500
      ZLAST = ZS(L)                                                     MX013600
      IF (TLAST .LE. 0.0.OR.ZLAST .LE. 0.0) GO TO 540                   MX013700
  550 L = L+1                                                           MX013800
      IF (L .GT. NLVLT) GO TO 590                                       MX013900
      IF (ZS(L) .LE. 0.0.OR.TS(L) .LE. 0.0) GO TO 550                   MX014000
      IF (ZS(L) .GT. HMLYR) GO TO 560                                   MX014100
      SUM1 = SUM1+(ZS(L)-ZLAST)*0.5*(TS(L)+TLAST)                       MX014200
      SUM2 = SUM2+(ZS(L)-ZLAST)                                         MX014300
      ZLAST = ZS(L)                                                     MX014400
      TLAST = TS(L)                                                     MX014500
      GO TO 550                                                         MX014600
  560 IF (SUM2 .LE. 0.0) GO TO 590                                      MX014700
      SUM1 = SUM1/SUM2                                                  MX014800
      SUM2 = 0.0                                                        MX014900
      L = L-1                                                           MX015000
  570 L = L+1                                                           MX015100
      IF (L .GT. NLVLT) GO TO 580                                       MX015200
```

```
      IF (TS(L) .LE. 0.0) GO TO 570
      SUM2 = AMAX1(SUM2,TS(L))
      GO TO 570
  580 IF (SUM2 .LE. 0.0) GO TO 590
      GN = 9.8*(1.0-SUM1/SUM2)
      IF (GN .LT. 0.1) GN = 0.1
      IF (GN .GT. 0.3) GN = 0.3
      GO TO 600
  590 GN = 0.1
  600 IF (UN .GT. -99.0.AND.VN .GT. -99.0) GO TO 620
      UN = UBRLYR
      VN = VBRLYR
      IF (UN .GT. -99.0.AND.VN .GT. -99.0) GO TO 620
      UN = SAVSPD
      VN = SAVDIR
      IF (UN .GT. -99.0.AND.VN .GT. -99.0) GO TO 620
  610 WRITE (XOTFIL,9001) UN,VN
  615 TYPE3 = -1
      GO TO 650
  620 IF (ABS(UN) .GT. 60.0.OR.ABS(VN) .GT. 60.0) GO TO 610
      IF (NETRDN .GE. -2.AND.NETRDN .LE. 4) GO TO 630
      WRITE (XOTFIL,9002) NETRDN
      GO TO 615
  630 CONTINUE
      IF (IFLAG1 .GT. 0.OR.IFLAG3 .GT. 0) GO TO 640
      WRITE (XOTFIL,9005)
      GO TO 615
  640 CONTINUE
      IF (GN .LE. -0.1) GN = 0.1
      WRITE (XOTFIL,9000) XSOURN,YSOURN,ZSOURN,UN,VN,PN,GN,ABLKN,DTLMDN,
     1ISMTHN,NCNTN,NETRDN,IFLG1
      IF (IFLAG4 .EQ. 1) WRITE (XOTFIL,9003)
      WRITE (XOTFIL,9004) IFLAG1,IFLAG2,IFLAG3
  650 CONTINUE
      RETURN
 9000 FORMAT ('1',32X,'*** ANALYSIS - INITIAL MIXING LAYER, SPEED CALCUL
     1ATIONS COMPLETE ***'//' COORDINATES OF RELEASE POINT (X,Y,Z) M (',
     2F10.2,',',F10.2,',',F10.2,')'/' VELOCITY COMPONENTS (U,V) MPS (',F
```

```
     38.4,',',F8.4,')'/' MIXING LAYER ELEVATION (P) M ',F8.2/' REDUCED GMX019100
     4RAVITY (G) M/S**2 ',F6.4/' MISC PARAMETERS (ABKLN,DTLMDN,ISMTHN,NCMX019200
     5NTN,NETRDN,IFLG1) (',F4.1,',',F4.2,',',I3,',',I3,',',I3,',',I2,')'MX019300
     6/)                                                                MX019400
9001 FORMAT ('0 *-*-* ERROR, U AND V COMPONENTS =',2E14.8,' NOT CORRECTMX019500
     1 CANNOT CONTINUE')                                                MX019600
9002 FORMAT ('0 *-*-* ERROR, NET RADIATION INDEX =',I8,' NOT CORRECT CAMX019700
     1NNOT CONTINUE')                                                   MX019800
9003 FORMAT (' MIXING LAYER ELEVATION DEFAULTED')                       MX019900
9004 FORMAT (' RAWINSONDE MODIFICATION FLAGS, IFLAG1=',I2,' IFLAG2=',I2MX020000
     1,' IFLAG3=',I2)                                                   MX020100
9005 FORMAT ('1 *-*-* ERROR - NO UPPER AIR OR SURFACE DATA. PROG CANNOTMX020200
     1CONTINUE'/)                                                       MX020300
9006 FORMAT ('1',24X,'*-* EPAMS ANALYSIS SOLUTION FOR YEAR/JULIAN/MONTHMX020400
     1/DAY/HOUR = ',I2,'/',I3,'/',I2,'/',I2,'/',I4,'Z *-*'/)            MX020500
9007 FORMAT (6I2)                                                       MX020600
     END                                                                MX020700
```

```
      SUBROUTINE MXUAOT                                                  MXU00100
C     THIS SUBROUTINE RETRIEVES THE UPPER AIR DATA                       MXU00200
      INCLUDE TIMES,LIST                                                 MXU00300
      INCLUDE NEWPRM,LIST                                                MXU00400
      INCLUDE OBUFER,LIST                                                MXU00500
      INCLUDE TDATA,LIST                                                 MXU00600
      INCLUDE WDATA,LIST                                                 MXU00700
      INCLUDE FILES,LIST                                                 MXU00800
      INCLUDE METPRM,LIST                                                MXU00900
      EQUIVALENCE (I1,I),(J1,J),(K1,K)                                   MXU01000
      INTEGER TIME,UABLK,OBUFRN                                          MXU01100
      DATA SAVSPD,SAVDIR/9999.,9999./                                    MXU01200
C     IPASS =1 IS LOOK FOR LESS THAN 3 HOURS                             MXU01300
C     IPASS =2 IS LOOK FOR LESS THAN 12 HOURS                            MXU01400
C     IPASS =3 IS NO HM FOUND COME BACK AND LOOK FOR OTHER DATA          MXU01500
      IF (IPASS .LT. 3) UABLK = MXUBLK+1                                 MXU01600
      GO TO (90,100,110),IPASS                                           MXU01700
C     SET HOUR INTERVAL MAXIMUM TO 3 HOURS                               MXU01800
   90 NUMHRS = 3                                                         MXU01900
      IFLAG1 = 1                                                         MXU02000
      GO TO 120                                                          MXU02100
C     SET HOUR INTERVAL MAXIMUM TO 12 HOURS                              MXU02200
  100 NUMHRS = 12                                                        MXU02300
      IFLAG1 = 2                                                         MXU02400
      GO TO 120                                                          MXU02500
  110 CONTINUE                                                           MXU02600
C     IPASS = 3 GOES HERE                                                MXU02700
  120 CONTINUE                                                           MXU02800
      UABLK = UABLK-1                                                    MXU02900
      IF (UABLK .LE. 0) GO TO 400                                        MXU03000
C     LOAD BUFFER WITH UPPER AIR DATA INVENTORY                          MXU03100
      CALL BLKIN(OBUFRN,UASINS,UABLK,OBFILE,ISTS)                        MXU03200
C     CALC TIME DIFF                                                     MXU03300
      J = RUNMAX                                                         MXU03400
  130 J = J-1                                                            MXU03500
      IF (J .LE. 0) GO TO 120                                            MXU03600
      IF (UATIME(J) .LE. 0) GO TO 130                                    MXU03700
      L = IDIFTM(TIME,UATIME(J))                                         MXU03800
```

```
      IF (L .LT. 0) GO TO 130                                           MXU03900
      IF (L .GT. NUMHRS) GO TO 400                                      MXU04000
C     DETERMINE INDICES OF ALL STATIONS WITHIN 100 KM                   MXU04100
      CALL CLSSTN(NSTATN,UASTNX,UASTNY,IDXSTA,STADST,STNMAX,100.)       MXU04200
      K = 0                                                             MXU04300
  170 K = K+1                                                           MXU04400
      IF (K .GT. NSTATN) GO TO 130                                      MXU04500
      I = IDXSTA(K)                                                     MXU04600
      IF (TLVLMX(I,J) .LE. 0.OR.WLVLMX(I,J) .LE. 0) GO TO 170           MXU04700
  320 IMAX = TLVLMX(I,J)                                                MXU04800
      KMAX = WLVLMX(I,J)                                                MXU04900
      CALL UADATA(I,J,IFLAG)                                            MXU05000
      M = FLD(35,1,IFLAG)                                               MXU05100
      L = FLD(34,1,IFLAG)                                               MXU05200
      IF (M .EQ. 0.AND.L .EQ. 0) GO TO 170                              MXU05300
C     PROVIDE DEFAULT SPD AND DIR IF NO OTHER DATA FOUND                MXU05400
      IF (ZS(1) .GT. 0.0) GO TO 350                                     MXU05500
      IF (ABS(US(1))+ABS(VS(1))) 355,355,350                            MXU05600
  350 SAVSPD = US(1)                                                    MXU05700
      SAVDIR = VS(1)                                                    MXU05800
      GO TO 365                                                         MXU05900
  355 IF (ZW(2) .GT. 0.0) GO TO 360                                     MXU06000
      IF (ABS(UW(2))+ABS(VW(2))) 365,365,360                            MXU06100
  360 SAVSPD = UW(2)                                                    MXU06200
      SAVDIR = VW(2)                                                    MXU06300
  365 CONTINUE                                                          MXU06400
C     FOUND DATA - EXAMINE FOR ENOUGH DATA TO CALC HM                   MXU06500
      IF (IMAX .LT. 3) GO TO 170                                        MXU06600
      M = 0                                                             MXU06700
      N850 = 0                                                          MXU06800
      N700 = 0                                                          MXU06900
      DO 380 L=1,IMAX                                                   MXU07000
      IF (ABS(PS(L)-85000.0) .GT. 100.0) GO TO 373                      MXU07100
      N850 = L                                                          MXU07200
  373 IF (ABS(PS(L)-70000.0) .GT. 100.0) GO TO 374                      MXU07300
      N700 = L                                                          MXU07400
  374 CONTINUE                                                          MXU07500
      IF (TV(L) .LE. 0.0.OR.ZS(L) .LE. 0.0) GO TO 380                   MXU07600
```

```
      IF (PS(L) .GT. 0.0.AND.PS(L) .LT. 70000.0) GO TO 381             MXU07700
      M = M+1                                                          MXU07800
  380 CONTINUE                                                         MXU07900
  381 IF (M .LT. 3) GO TO 170                                          MXU08000
      JUTIME = UATIME(J)                                               MXU08100
      NLVLT = IMAX                                                     MXU08200
      STATNZ = UASTNZ(I)                                               MXU08300
      STATNX = UASTNX(I)                                               MXU08400
      STATNY = UASTNY(I)                                               MXU08500
      DO 390 L=1,25                                                    MXU08600
      IF (L .GT. NLVLT) GO TO 390                                      MXU08700
      IF (POTMPN(L) .LE. 0.0) POTMPN(L) = TA(L)                        MXU08800
      IF (ZELEVN(L) .LE. 0.0) ZELEVN(L) = ZS(L)                        MXU08900
  390 CONTINUE                                                         MXU09000
      GO TO 500                                                        MXU09100
  400 NSTATN = 0                                                       MXU09200
  410 IFLAG1 = 0                                                       MXU09300
      N850 = 0                                                         MXU09400
      N700 = 0                                                         MXU09500
  500 CONTINUE                                                         MXU09600
C*C                                                                    MXU09700
      IF (KFLG2 .EQ. 0) GO TO 630                                      MXU09800
      WRITE (XOTFIL,9000)                                              MXU09900
      IF (IFLAG1 .EQ. 0) GO TO 620                                     MXU10000
      WRITE (XOTFIL,9001) UABLK,J1,K1,UASTNS(I),STATNX,STATNY,STATNZ,  MXU10100
     1TIME,JUTIME                                                      MXU10200
      WRITE (XOTFIL,9002)                                              MXU10300
      DO 600 M=1,NLVLT                                                 MXU10400
  600 WRITE (XOTFIL,9003) M,PS(M),QS(M),TS(M),ZS(M),TV(M),TA(M),US(M), MXU10500
     1VS(M),DELTA(M),PD(M)                                             MXU10600
      WRITE (XOTFIL,9004)                                              MXU10700
      DO 610 M=1,KMAX                                                  MXU10800
  610 WRITE (XOTFIL,9005) M,UW(M),VW(M),ZW(M)                          MXU10900
      GO TO 630                                                        MXU11000
  620 WRITE (XOTFIL,9006) NUMHRS                                       MXU11100
  630 CONTINUE                                                         MXU11200
 9000 FORMAT ('1 *** UPPER AIR DATA PROCESSED ***')                    MXU11300
 9001 FORMAT ('0BLOCK=',I3,' RUN INDEX=',I2,' STA INDEX=',I3,' STATION=',MXU11400
```

```
     1,I6,' UTM X=',F8.3,' UTM Y=',F9.3,' UTM Z=',F9.3,' CALC FOR TIME='MXU11500
     2,I6/' OBS TIME=',I6)                                               MXU11600
9002 FORMAT ('0 UPPER AIR OBS. PRESS. DATA'/' LEVEL  PRESS.    MIXING   MXU11700
     1  TEMP   ELEV  VIRT-TEMP POT-TEMP U-SPD COMP V-SPD COMP HYDRO EQU MXU11800
     2EXPO DEW PNT DEP'/6X,'(PASCALS)  RATIO    (DEG K)   (M)   (DEG K) MXU11900
     3  (DEG K)     (M/S)       (M/S)',21X,'(DEG K)')                   MXU12000
9003 FORMAT (2X,I2,F10.1,E11.6,F7.2,F9.2,F8.2,F9.2,F10.2,F11.2,E15.6,F1MXU12100
     11.2)                                                              MXU12200
9004 FORMAT ('0 UPPER AIR OBS. WIND DATA'/' LEVEL U-SPD COMP V-SPD COMPMXU12300
     1 ELEV'/10X,'(M/S)',6X,'(M/S)',4X,'(M)')                           MXU12400
9005 FORMAT (2X,I2,2F11.2,F10.2)                                        MXU12500
9006 FORMAT ('0 NO UPPER AIR DATA FOUND WITHIN ',I4,' HOURS, PROG CONTIMXU12600
     1NUES')                                                            MXU12700
C*C                                                                     MXU12800
      RETURN                                                            MXU12900
      END                                                               MXU13000
```

```
      SUBROUTINE MXGWDT(PU,TU)
      INCLUDE TIMES,LIST
      INCLUDE NEWPRM,LIST
      INCLUDE FILES,LIST
      INCLUDE GWDATA,LIST
      INCLUDE METPRM,LIST
      INCLUDE ABUFER,LIST
      INCLUDE OBUFER,LIST
      INCLUDE TDATA,LIST
      DIMENSION PU(1),TU(1)
      INTEGER FLAG,TIME
      INTEGER OBUFRN,GWCBLK
      IFLAG2 = 0
      GWCBLK = MXGBLK+1
   90 GWCBLK = GWCBLK-1
      IF (GWCBLK .LE. 0) GO TO 700
C     OBTAIN GWC PREDICTION DATA INVENTORY
      CALL BLKIN(OBUFRN,GRIDX(1),GWCBLK,GWCPFL,ISTS)
C     FIND CLOSEST GWC STATION
      CALL CLSSTN(NSTATN,GRIDX,GRIDY,IDXSTA,STADST,GRDMAX,100.0)
      IF (NSTATN .NE. 0) GO TO 200
      GO TO 90
  200 L = 0
  205 L = L+1
      IF (L .GT. NSTATN) GO TO 90
      I = IDXSTA(L)
      K = RUNMAX+1
  210 K = K-1
      IF (K .LT. 1) GO TO 205
      IF (ANLTIM(K) .LE. 0) GO TO 210
      IF (ANLTIM(K) .GT. JUTIME) GO TO 210
      M = IDIFTM(TIME,ANLTIM(K))
      IF (M .GT. 18) GO TO 205
      GTIME = TIME
      IFLAG2 = 0
      CALL GWCDTA(I,GRDBLK(K),FLAG)
      IF (FLAG .EQ. 0) GO TO 205
      IFLAG2 = 1
```

```
      IF (N850 .EQ. 0.OR.N700 .EQ. 0) GO TO 700                          MXG03900
      IF (T(3) .LE. 0.0.OR.TS(N700) .LE. 0.0) GO TO 700                  MXG04000
      T850 = T(3)-TS(N700)                                               MXG04100
      IF (ABS(T850) .GT. 5.0.OR.Q(3) .LE. 0.0) GO TO 700                 MXG04200
      TS(N700) = T(3)                                                    MXG04300
      TV(N700) = (1.0+0.61*(Q(3)/(1.0-Q(3))))*T(3)                       MXG04400
      US(N850) = U(2)                                                    MXG04500
      VS(N850) = V(2)                                                    MXG04600
      US(N700) = U(3)                                                    MXG04700
      VS(N700) = V(3)                                                    MXG04800
      IFLAG2 = 2                                                         MXG04900
      IF (T850 .GT. 0.0) IFLAG2 = 3                                      MXG05000
      GO TO 800                                                          MXG05100
  700 CONTINUE                                                           MXG05200
C     IFLAG2 = 0 NO DATA                                                 MXG05300
  800 CONTINUE                                                           MXG05400
      IF (KFLG2 .EQ. 0) GO TO 820                                        MXG05500
      WRITE (XOTFIL,9000)                                                MXG05600
      IF (IFLAG2 .GT. 0) GO TO 810                                       MXG05700
      WRITE (XOTFIL,9001)                                                MXG05800
      GO TO 820                                                          MXG05900
  810 WRITE (XOTFIL,9002) GWCBLK,K,I,GRIDX(I),GRIDY(I),TIME,ANLTIM(K),   MXG06000
     1GTIME                                                              MXG06100
      IF (IFLAG2 .GT. 1) WRITE (XOTFIL,9003) T850,T(3)                   MXG06200
      WRITE (XOTFIL,9004)                                                MXG06300
      DO 815 I=1,4                                                       MXG06400
  815 WRITE (XOTFIL,9005) I,P(I),Q(I),T(I),U(I),V(I),Z(I)                MXG06500
  820 CONTINUE                                                           MXG06600
 9000 FORMAT ('1 *** GWC PREDICTION DATA PROCESSED ***')                 MXG06700
 9001 FORMAT ('0 NO GWC DATA FOUND OR DATA DID NOT MEET RAOB MODIFICATIOMXG06800
     1N SPECS')                                                          MXG06900
 9002 FORMAT ('0 BLOCK=',I3,' RUN INDEX=',I2,' STA INDEX=',I3,' UTM X=',MXG07000
     1F8.3,' UTM Y=',F9.3,' CALC FOR TIME=',I6,' UPPER AIR OBS TIME=',I6MXG07100
     2/' GWC PRE TIME=',I6)                                              MXG07200
 9003 FORMAT (' DELTA T=',F8.3,' T AT 700=',F8.3,' RAWIN MODIFICATIONS MMXG07300
     1ADE')                                                              MXG07400
 9004 FORMAT ('0 GWC PREDICTION DATA'/' LEVEL  PRESS.  SPECIFIC  TEMP.  MXG07500
     1U-SPD COMP V-SPD COMP   ELEV.'/6X,'(PASCALS)   HUM.   (DEG K)    (MXG07600
```

```
77*         2M/S)       (M/S)       (M)')                                    MXG07700
78*    9005 FORMAT (2X,I2,F10.1,E10.4,F8.2,F9.2,2F11.2)                      MXG07800
79*         RETURN                                                           MXG07900
80*         END                                                              MXG08000
```

```
      SUBROUTINE MXSFDT                                                 MXS00100
      INCLUDE FILES,LIST                                                MXS00200
      INCLUDE TIMES,LIST                                                MXS00300
      INCLUDE TDATA,LIST                                                MXS00400
      INCLUDE WDATA,LIST                                                MXS00500
      INCLUDE NEWPRM,LIST                                               MXS00600
      INCLUDE SFDATA,LIST                                               MXS00700
      INCLUDE METPRM,LIST                                               MXS00800
      INCLUDE SBUFER,LIST                                               MXS00900
      INCLUDE ABUFER,LIST                                               MXS01000
      INCLUDE GWDATA,LIST                                               MXS01100
      DIMENSION JDXSTA(25),STBDST(25),SOLAR(3)                          MXS01200
      DIMENSION PTTUS(40,8)                                             MXS01300
      EQUIVALENCE (PTTUS,PS)                                            MXS01400
      REAL LAT,LON                                                      MXS01500
      LOGICAL CEILNG                                                    MXS01600
      INTEGER FLAG,TIME                                                 MXS01700
      INTEGER OBUFRN,SFCBLK                                             MXS01800
      DATA SOLAR/60.,35.,15./                                           MXS01900
C     THIS SUBROUTINE EXAMINES THE SURFACE DATA TO DETERMINE IF A       MXS02000
C     MODIFICATION OF THE RAWINSONDE DATA IS DESIRED AND ALSO TO        MXS02100
C     CALCULATE THE NET RADIATION INDEX                                 MXS02200
      ISW1 = 0                                                          MXS02300
      IFLAG3 = 0                                                        MXS02400
      NUMHRS = 3                                                        MXS02500
      NSTATN = 0                                                        MXS02600
C     IF NO STATIONS WITHIN 3 HRS OLD DEFAULT NET RADIATION             MXS02700
      SFCBLK = MXSBLK+1                                                 MXS02800
  150 SFCBLK = SFCBLK-1                                                 MXS02900
      IF (SFCBLK .LE. 0) GO TO 350                                      MXS03000
C     GET SURFACE DATA INVENTORY                                        MXS03100
      CALL BLKIN(OBUFRN,SFCSTN,SFCBLK,SFCOBF,ISTS)                      MXS03200
C     DETERMINE CLOSEST STATION WITHIN 50 KM                            MXS03300
      CALL CLSSTN(NSTATN,SFSTNX,SFSTNY,JDXSTA,STBDST,STNMAX,50.0)       MXS03400
C     IF NO STATIONS TRY FOR NEXT BLOCK                                 MXS03500
      L = 0                                                             MXS03600
  160 L = L+1                                                           MXS03700
      IF (L .GT. NSTATN) GO TO 150                                      MXS03800
```

```
      J = RUNMAX                                                        MXS03900
  250 J = J-1                                                           MXS04000
      IFLAG3 = 0                                                        MXS04100
      IF (J .LE. 0) GO TO 160                                           MXS04200
      IF (RUNTIM(J) .LE. 0) GO TO 250                                   MXS04300
      N = IDIFTM(TIME,RUNTIM(J)/100)                                    MXS04400
      IF (N .LT. 0) GO TO 250                                           MXS04500
      IF (J .EQ. RUNMAX.AND.N .GT. NUMHRS) GO TO 350                    MXS04600
      IF (N .GT. NUMHRS) GO TO 160                                      MXS04700
      I = JDXSTA(L)                                                     MXS04800
      STIME = RUNTIM(J)                                                 MXS04900
      CALL BLKIN(5500,A(1),SFCBLK,SBFILE,ISTS)                          MXS05000
      CALL SFCDTA(I,J,FLAG)                                             MXS05100
      IF (FLAG .EQ. 0) GO TO 250                                        MXS05200
C     PROVIDE FOR DEFAULT SPD AND DIR IF NO THERE                       MXS05300
      IF (WD .GT. 360.0) WD = -1.0                                      MXS05400
      IF (WD .GE. 0.0) WD = WD+SFSTNA(I)                                MXS05500
      IF (WD .LT. 0.0.OR.WS .LT. 0.0) GO TO 330                         MXS05600
      SAVDIR = (WD+180.0)*.01745329                                     MXS05700
      SAVSPD = WS*SIN(SAVDIR)                                           MXS05800
      SAVDIR = WS*COS(SAVDIR)                                           MXS05900
  330 IF (ISW1 .EQ. 2) GO TO 700                                        MXS06000
C     SET LOCATION OF STATION                                           MXS06100
      XST1 = SFSTNX(I)*1.0E3                                            MXS06200
      YST1 = SFSTNY(I)*1.0E3                                            MXS06300
      SFSZ = SFSTNZ(I)                                                  MXS06400
      IF (SURFTN .LE. 0.0.AND.TSS .GT. 0.0) SURFTN = TSS                MXS06500
C     SET STATION FOUND FLAG ON                                         MXS06600
      ISW2 = 1                                                          MXS06700
      GO TO 400                                                         MXS06800
C     DEFAULT LOCATION FOR NET RADIATION CALCULATION                    MXS06900
  350 IF (ISW1 .GT. 0) GO TO 700                                        MXS07000
      XST1 = XSOURN                                                     MXS07100
      YST1 = YSOURN                                                     MXS07200
      IF (IFLAG3 .EQ. 0) SFSZ = ZSOURN                                  MXS07300
C     TURN STATION FOUND FLAG OFF                                       MXS07400
      ISW2 = 0                                                          MXS07500
C     CALCULATE THE NET RADIATION                                       MXS07600
```

```
  400 CONTINUE                                                                  MXS07700
C     CALCULATE THE NET RADIATION INDEX - NETRDN                                MXS07800
C     FIRST CALC THE LONGITUDE AND LATITUDE                                     MXS07900
      CALL LONLAT(XST1,YST1,LON,LAT)                                            MXS08000
C     USE DEFAULT IF ISW2 = 0                                                   MXS08100
      IF (ISW2 .EQ. 0) GO TO 600                                                MXS08200
      NOBS = FLAG/16                                                            MXS08300
      K = FLD(32,4,FLAG)                                                        MXS08400
      FLAG = K+1                                                                MXS08500
C     DETERMINE CLOUD COVER AND CEILNG- CLDCVR IN TENTHS, CEILNG IF <2KMMXS08600
C     THEN TRUE, ELSE FALSE                                                     MXS08700
C     FLAG GIVES TYPE OF SURFACE DATA                                           MXS08800
C     FLAG = 1 SYNOPTIC                                                         MXS08900
C     FLAG = 2 AIRWAYS                                                          MXS09000
C     FLAG = 3 METAR                                                            MXS09100
C     FLAG = 4 SYNOPTIC-AIRWAYS MERGED                                          MXS09200
C     FLAG = 5 SYNOPTIC-METAR MERGED                                            MXS09300
C     FLAG = 6 RESERVED                                                         MXS09400
      GO TO (500,540,560,500,560,560),FLAG                                      MXS09500
C     SYNOPTIC OR MERGED SYNOPTIC                                               MXS09600
  500 IF (TSC .LT. 0 .OR. TSC .GT. 9) GO TO 520                                 MXS09700
      CLSDVR = TSC                                                              MXS09800
  510 CLDCVR = CLDCVR*0.125                                                     MXS09900
      GO TO 550                                                                 MXS10000
  520 IF (NH .LT. 0 .OR. NH .GT. 9) GO TO 530                                   MXS10100
      CLDCVR = NH                                                               MXS10200
      GO TO 510                                                                 MXS10300
  530 IF (NS .LT. 0 .OR. NS .GT. 9) GO TO 560                                   MXS10400
      CLDCVR = NS                                                               MXS10500
      GO TO 510                                                                 MXS10600
C     AIRWAYS                                                                   MXS10700
  540 IF (IABS(TSC) .GT. 10.OR.TSC .EQ. -1) GO TO 530                           MXS10800
      CLDCVR = IABS(TSC)*0.1                                                    MXS10900
C     IF CEILNG < 2 KM SET TRUE, ELSE SET FALSE                                 MXS11000
  550 IF (HSHS .LT. 0 .OR. HSHS .GT. 99) GO TO 560                              MXS11100
      IF (HSHS .LT. 56 .AND. HSHS .GT. 50) GO TO 560                            MXS11200
      CEILNG = .TRUE.                                                           MXS11300
      IF (HSHS .GE. 57.AND.HSHS .LT. 90) CEILNG = .FALSE.                       MXS11400
```

```
          IF (HSHS .GT. 97) CEILNG = .FALSE.                          MXS11500
          GO TO 600                                                   MXS11600
      560 ISW2 = 0                                                    MXS11700
C         CALCULATE 1 HOUR PAST SUNRISE TO 1 HOUR BEFORE SUNSET       MXS11800
      600 CALL DAYNIT(LON,LAT,SNRISE,SUNSET,ALPHA)                    MXS11900
C         DETERMINE THE PRESENT HOUR                                  MXS12000
          TIMES = HOUR+FLOAT(MINUTE)*1.6666666E-2                     MXS12100
          NETRDM = -1                                                 MXS12200
C         DETERMINE IF DAT OR NIGHT                                   MXS12300
          IF (SNRISE .GT. SUNSET) GO TO 604                           MXS12400
          IF (TIMES .GE. SNRISE.AND.TIMES .LE. SUNSET) GO TO 630      MXS12500
          GO TO 605                                                   MXS12600
      604 IF (TIMES .GE. SNRISE.OR.TIMES .LE. SUNSET) GO TO 630       MXS12700
      605 CONTINUE                                                    MXS12800
C         NIGHT TIME                                                  MXS12900
          IF (ISW2 .EQ. 0) GO TO 610                                  MXS13000
          IF (CLDCVR .LE. 0.4) GO TO 620                              MXS13100
      610 NETRDN = -1                                                 MXS13200
          GO TO 670                                                   MXS13300
      620 NETRDN = -2                                                 MXS13400
          GO TO 670                                                   MXS13500
C         DAY TIME                                                    MXS13600
      630 CONTINUE                                                    MXS13700
          IF (ALPHA .GT. 90.0) ALPHA = ABS(ALPHA-180.0)               MXS13800
          DO 640 NETRDN=1,3                                           MXS13900
          IF (ALPHA .GT. SOLAR(NETRDN)) GO TO 650                     MXS14000
      640 CONTINUE                                                    MXS14100
          NETRDN = 4                                                  MXS14200
      650 NETRDN = 5-NETRDN                                           MXS14300
          NETRDM = NETRDN-1                                           MXS14400
          IF (ISW2 .EQ. 1) GO TO 660                                  MXS14500
          NETRDN = NETRDN-1                                           MXS14600
          GO TO 670                                                   MXS14700
      660 IF (CLDCVR .LE. 0.5) GO TO 670                              MXS14800
          NETRDN = NETRDN-1                                           MXS14900
          IF (CEILNG) NETRDN = NETRDN-1                               MXS15000
          IF (NETRDN .LT. 1) NETRDN = 1                               MXS15100
      670 CONTINUE                                                    MXS15200
```

```
      ISW1 = 1                                                          MXS15300
      IF (ISW2 .EQ. 1) ISW1 = 2                                         MXS15400
  700 IF (N .GT. NUMHRS) GO TO 800                                      MXS15500
C     CALCULATE THE SURFACE PRESSURE AT THE STATION                     MXS15600
      IF (AS .LE. 0.0) GO TO 710                                        MXS15700
      SLP = AS*(1.0-.0065*SFSTNZ(I)/288.0)**5.2658*1013.25/29.92        MXS15800
      GO TO 720                                                         MXS15900
  710 IF (SLP .GT. 0.0.AND.TSS .GT. 0.0) GO TO 715                      MXS16000
C     MISSING DATA GO BACK IF NEEDED                                    MXS16100
      IF (ISW2 .EQ. 0.OR.IFLAG1 .GT. 1) GO TO 250                       MXS16200
      GO TO 800                                                         MXS16300
  715 CONTINUE                                                          MXS16400
      SLP = EXP(ALOG(SLP)-SFSTNZ(I)/(29.29*TSS))                        MXS16500
  720 CONTINUE                                                          MXS16600
      IFLAG3 = 1                                                        MXS16700
      IF (IFLAG1 .LE. 1) GO TO 800                                      MXS16800
      IF (ABS(SLP-PS(1)*.01) .GT. 10.0) GO TO 250                       MXS16900
      IF (TD .LE. 0.0) GO TO 250                                        MXS17000
      TS(1) = TSS                                                       MXS17100
      T850 = 1.0-373.16/(TSS-TD)                                        MXS17200
      T850 = 1013.25*EXP(T850*(13.3185+T850*(-1.976+T850*(-.6445-.1299* MXS17300
     1T850))))                                                          MXS17400
      T850 = .622*T850/(PS(1)*.01-T850)                                 MXS17500
      TV(1) = (1.0+0.61*T850)*TSS                                       MXS17600
      IF (WD .LT. 0.0.OR.WS .LT. 0.0) GO TO 725                         MXS17700
      VS(1) = (WD+180.0)*.0174532925                                    MXS17800
      US(1) = WS*SIN(VS(1))                                             MXS17900
      VS(1) = WS*COS(VS(1))                                             MXS18000
  725 CONTINUE                                                          MXS18100
      IFLAG3 = 2                                                        MXS18200
      IF (IFLAG2 .LT. 2) GO TO 730                                      MXS18300
C     GWC MODS MADE                                                     MXS18400
      IF (T(2) .LE. 0.0) GO TO 800                                      MXS18500
      T850 = T(2)-TSS                                                   MXS18600
      IF (ABS(T850) .GT. 5.0) GO TO 730                                 MXS18700
      IF (Q(2) .LE. 0.0) GO TO 800                                      MXS18800
      IFLAG3 = 3                                                        MXS18900
      IF (T850 .GT. 0.0) IFLAG3 = 4                                     MXS19000
```

```
      TS(N850) = T(2)                                               MXS19100
      TV(N850) = (1.0+0.61*Q(2)/(1.0-Q(2)))*T(2)                    MXS19200
      US(N850) = U(2)                                               MXS19300
      VS(N850) = V(2)                                               MXS19400
      GO TO 740                                                     MXS19500
  730 IF (TS(N850) .LE. 0.0) GO TO 800                              MXS19600
      T850 = TS(N850)-TSS                                           MXS19700
      IF (ABS(T850) .LE. 5.0) GO TO 740                             MXS19800
      IFLAG1 = 0                                                    MXS19900
      GO TO 800                                                     MXS20000
C     REMOVE LEVELS BETWEEN 850 MB AND SURFACE                      MXS20100
  740 DO 741 L=1,8                                                  MXS20200
  741 PTTUS(2,L) = PTTUS(N850,L)                                    MXS20300
      N850 = 2                                                      MXS20400
      N = 2                                                         MXS20500
      M = N850+1                                                    MXS20600
      IF (M .GT. NLVLT) GO TO 781                                   MXS20700
C     REMOVE LEVELS BETWEEN 850 AND 800 MB                          MXS20800
      DO 755 L=M,NLVLT                                              MXS20900
      IF (PS(L) .GT. 79900.0) GO TO 755                             MXS21000
      N = N+1                                                       MXS21100
      DO 750 K=1,8                                                  MXS21200
  750 PTTUS(N,K) = PTTUS(L,K)                                       MXS21300
      IF (ABS(PS(L)-70000.0) .LE. 100.0) N700 = N                   MXS21400
  755 CONTINUE                                                      MXS21500
      NLVLT = N                                                     MXS21600
      IF (IFLAG2 .EQ. 3) GO TO 780                                  MXS21700
      IF (IFLAG3 .EQ. 4) GO TO 781                                  MXS21800
C     REMOVE LEVELS BETWEEN 850 AND 700 MB                          MXS21900
  765 N = N850                                                      MXS22000
      DO 766 K=N700,NLVLT                                           MXS22100
      N = N+1                                                       MXS22200
      DO 766 L=1,8                                                  MXS22300
  766 PTTUS(N,L) = PTTUS(K,L)                                       MXS22400
      NLVLT = N                                                     MXS22500
      GO TO 781                                                     MXS22600
  780 IF (IFLAG3 .EQ. 4) GO TO 765                                  MXS22700
  781 L = 0                                                         MXS22800
```

```
  782 L = L+1                                                          MXS22900
      IF (L .GT. NLVLT) GO TO 800                                      MXS23000
      IF (ZS(L) .GE. ZS(L-1)) GO TO 782                                MXS23100
      M = L+1                                                          MXS23200
      IF (M .GT. NLVLT) GO TO 786                                      MXS23300
      DO 785 K=M,NLVLT                                                 MXS23400
      DO 785 N=1,8                                                     MXS23500
  785 PTTUS(K-1,N) = PTTUS(K,N)                                        MXS23600
      L = L-1                                                          MXS23700
  786 NLVLT = NLVLT-1                                                  MXS23800
      GO TO 782                                                        MXS23900
  800 CONTINUE                                                         MXS24000
C*C                                                                    MXS24100
      IF (KFLG2 .EQ. 0) GO TO 850                                      MXS24200
      WRITE (XOTFIL,9000)                                              MXS24300
      IF (IFLAG3 .EQ. 0) GO TO 840                                     MXS24400
      WRITE (XOTFIL,9001) SFCBLK,J,I,SFCSTN(I),SFSTNX(I),SFSTNY(I),    MXS24500
     1SFSTNZ(I),STIME                                                  MXS24600
      WRITE (XOTFIL,9005) WD,WS,WG,SLP,BT,TSS,TD,AS,PA,TSC,PW,VV,WW1,WW2MXS24700
     1,WW3,WW4,NH,CL,H,CM,CH,NS,K,CT,HSHS                              MXS24800
      IF (IFLAG3 .GE. 2) WRITE (XOTFIL,9002)                           MXS24900
  840 IF (IFLAG3 .EQ. 0) WRITE (XOTFIL,9003)                           MXS25000
      WRITE (XOTFIL,9004) NETRDN,ALPHA,SNRISE,SUNSET,XST1,YST1,LON,LAT, MXS25100
     1SFSZ                                                             MXS25200
  850 CONTINUE                                                         MXS25300
 9000 FORMAT ('1 *** SURFACE DATA PROCESSED ***')                      MXS25400
 9001 FORMAT ('0 BLOCK=',I4,' RUN INDEX=',I3,' STA INDEX=',I3,' STATION=MXS25500
     1',I6,' UTM X=',F8.3,' UTM Y=',F9.3,' UTM Z=',F8.3,' SURF OBS TIME=MXS25600
     2',I8)                                                            MXS25700
 9002 FORMAT (' RAWIN MODIFICATIONS MADE')                             MXS25800
 9003 FORMAT ('0 NO SURFACE DATA FOUND, DEFAULT NET RADIATION INDEX')  MXS25900
 9004 FORMAT ('0 NET RAD INDEX=',I3,' SOL ALT ANG=',F8.3,' SUN RISE +1HRMXS26000
     1=',F9.3,' SUN SET -1HR=',F9.3,' UTM X=',F10.2/'  UTM Y=',F11.2,' LMXS26100
     2ONGITUDE=',F9.4,' LATITUDE=',F9.4,' SURFACE Z=',F9.3)            MXS26200
 9005 FORMAT ('0SURFACE DATA, DIR (DEG)=',F7.2,', SPD (M/S)=',F7.2,', WIMXS26300
     1ND GUSTS (M/S)=',F7.2,', PRESS (MB)=',F7.2,', BAROMETRIC TEND.=',IMXS26400
     28/' TEMP (DEG K)=',F7.2,', DEW PNT DEP (DEG K)=',F7.2,', ALTIM SETMXS26500
     3TING (IN HG)=',F6.2,', 6HR PRECIP (MM)=',F6.2,', TOT SKY COVER (8TMXS26600
```

```
     4HS)=',I4/' PAST WEATHER (WMO 4500)=',I6,', VIS (M)=',I6,', PRES WEMXS26700
     5A WWI I=1,4 (WMO 4677)=',4I5,', LOW CLDS (8THS)=',I4/' TYPE LOW CLMXS26800
     6DS (WMO 513)=',I4,', HT LOW CLD (WMO 1600)=',I6,', TYPE MID CLD (WMXS26900
     7MO 515)=',I4,', TYPE HI CLD (WMO 509)=',I4/' AMT CLDS (8THS)=',I4,MXS27000
     8', CLASSIF CLDS=',I4,', TYPE CLD IN HT CLASS=',I4,', HT TO BASE CLMXS27100
     9DS (WMO 1677)=',I6)                                                MXS27200
C*C                                                                      MXS27300
      RETURN                                                             MXS27400
      END                                                                MXS27500
```

```
      SUBROUTINE LYRAHM(UBRLYR,VBRLYR,HMLYR)                            LYR00100
      INCLUDE FILES,LIST                                                LYR00200
      INCLUDE NEWPRM,LIST                                               LYR00300
      INCLUDE TDATA,LIST                                                LYR00400
      INCLUDE WDATA,LIST                                                LYR00500
      INCLUDE METPRM,LIST                                               LYR00600
      INCLUDE SFDATA,LIST                                               LYR00700
      INCLUDE GWDATA,LIST                                               LYR00800
      REAL MINDPT                                                       LYR00900
      EQUIVALENCE (STATHT,STATNZ)                                       LYR01000
      DATA DPDZMN/-.0005/,MINDPT/100.0/,HMMAX/3000.0/,HMMIN/30.0/,      LYR01100
     1DMISS/0.0/                                                        LYR01200
      ISW = 0                                                           LYR01300
      HMLYR = -1.0                                                      LYR01400
      IF (IFLAG1 .EQ. 0) GO TO 860                                      LYR01500
      I = 0                                                             LYR01600
  350 I = I+1                                                           LYR01700
      IF (I .GT. NLVLT) GO TO 860                                       LYR01800
      IF (PS(I) .GT. 0.0.AND.PS(I) .LT. 70000.0) GO TO 860              LYR01900
      IF (ZS(I) .LE. 0.0.OR.TV(I) .LE. 0.0) GO TO 350                   LYR02000
      TLAST = TV(I)                                                     LYR02100
      ZLAST = ZS(I)                                                     LYR02200
      ZBASE = STATHT                                                    LYR02300
      HM = HMMAX+ZBASE                                                  LYR02400
      BASE = 0.0                                                        LYR02500
      HB = 0.0                                                          LYR02600
      ISW1 = 0                                                          LYR02700
      IB = 0                                                            LYR02800
  490 I = I+1                                                           LYR02900
      IF (I .GT. NLVLT) GO TO 540                                       LYR03000
      IF (PS(I) .GT. 0.0.AND.PS(I) .LT. 70000.0) GO TO 540              LYR03100
      IF (TV(I) .LE. DMISS.OR.ZS(I) .LE. DMISS) GO TO 490               LYR03200
      TDIFF = TV(I)-TLAST                                               LYR03300
      ZDIFF = ZS(I)-ZLAST                                               LYR03400
      IF (ZDIFF .LE. 0.0) GO TO 490                                     LYR03500
      DPDZ = TDIFF/ZDIFF                                                LYR03600
      IF (DPDZ .LT. DPDZMN) GO TO 510                                   LYR03700
  500 HB = HB+ZDIFF                                                     LYR03800
```

```
      IB = IB+1                                                         LYR03900
      IF (IB .NE. 1) GO TO 505                                          LYR04000
      BASE = ZLAST                                                      LYR04100
      TSAV = TLAST                                                      LYR04200
  505 CONTINUE                                                          LYR04300
      IF (HB .GE. MINDPT) GO TO 530                                     LYR04400
      ISW1 = 1                                                          LYR04500
      GO TO 520                                                         LYR04600
  510 HB = 0.0                                                          LYR04700
      IB = 0                                                            LYR04800
      IF (ISW1 .NE. 1) GO TO 515                                        LYR04900
      I = I-1                                                           LYR05000
      ISW1 = 0                                                          LYR05100
      ZLAST = BASE                                                      LYR05200
      TLAST = TSAV                                                      LYR05300
      BASE = 0.0                                                        LYR05400
      GO TO 490                                                         LYR05500
  515 BASE = 0.0                                                        LYR05600
  520 ZLAST = ZS(I)                                                     LYR05700
      TLAST = TV(I)                                                     LYR05800
      GO TO 490                                                         LYR05900
  530 HM = BASE                                                         LYR06000
      IF (HM .LT. HMMIN+ZBASE) HM = HMMIN+ZBASE                         LYR06100
      IF (HM .GT. HMMAX+ZBASE) HM = HMMAX+ZBASE                         LYR06200
  540 HMLYR = HM                                                        LYR06300
      ISW = 1                                                           LYR06400
C     CALC LAYER SPD AND DIR                                            LYR06500
      ZLAST = HMLYR                                                     LYR06600
      SUM1 = 0.0                                                        LYR06700
      SUM2 = 0.0                                                        LYR06800
      SUM3 = 0.0                                                        LYR06900
      ILVL = 0                                                          LYR07000
      ZLST1 = ZBASE                                                     LYR07100
      IB = 0                                                            LYR07200
      J = 0                                                             LYR07300
  800 J = J+1                                                           LYR07400
      IF (ZLST1 .GE. ZLAST) GO TO 850                                   LYR07500
      IF (J .GT. NLVLT) GO TO 850                                       LYR07600
```

```
      IF (ZS(J) .LE. DMISS) GO TO 800                                   LYR07700
      IF (ZS(J) .LT. ZLST1) GO TO 800                                   LYR07800
      IF (IB .EQ. 0) GO TO 830                                          LYR07900
      IF (ZS(J) .LE. ZLAST) GO TO 810                                   LYR08000
C     INTERP                                                            LYR08100
      UB = US(J)-(ZS(J)-ZLAST)*(US(J)-ULST)/(ZS(J)-ZLST1)               LYR08200
      VB = VS(J)-(ZS(J)-ZLAST)*(VS(J)-VLST)/(ZS(J)-ZLST1)               LYR08300
      HT = ZLAST                                                        LYR08400
      GO TO 820                                                         LYR08500
  810 UB = US(J)                                                        LYR08600
      VB = VS(J)                                                        LYR08700
      HT = ZS(J)                                                        LYR08800
  820 SUM1 = SUM1+(HT-ZLST1)*0.5*(UB+ULST)                              LYR08900
      SUM2 = SUM2+(HT-ZLST1)*0.5*(VB+VLST)                              LYR09000
      SUM3 = SUM3+(HT-ZLST1)                                            LYR09100
      GO TO 840                                                         LYR09200
  830 UB = US(J)                                                        LYR09300
      VB = VS(J)                                                        LYR09400
      HT = ZS(J)                                                        LYR09500
      IB = 1                                                            LYR09600
  840 ULST = UB                                                         LYR09700
      VLST = VB                                                         LYR09800
      ZLST1 = HT                                                        LYR09900
      GO TO 800                                                         LYR10000
  850 IF (SUM3 .LE. 0.0) GO TO 860                                      LYR10100
      UBRLYR = SUM1/SUM3                                                LYR10200
      VBRLYR = SUM2/SUM3                                                LYR10300
      GO TO 900                                                         LYR10400
  860 CONTINUE                                                          LYR10500
  900 CONTINUE                                                          LYR10600
      IF (KFLG2 .EQ. 0) GO TO 921                                       LYR10700
      IF (ISW .EQ. 0) GO TO 910                                         LYR10800
      WRITE (XOTFIL,9000) HMLYR                                         LYR10900
      GO TO 911                                                         LYR11000
  910 WRITE (XOTFIL,9001)                                               LYR11100
  911 WRITE (XOTFIL,9002)                                               LYR11200
      DO 920 I=1,NLVLT                                                  LYR11300
      WRITE (XOTFIL,9003) I,PS(I),TS(I),ZS(I),TV(I),US(I),VS(I)         LYR11400
```

```
  920 CONTINUE                                                          LYR11500
  921 CONTINUE                                                          LYR11600
 9000 FORMAT ('0',5X,'*-* CALCULATED HM =',F9.3)                        LYR11700
 9001 FORMAT ('0',5X,'*-* HM NOT CALCULATED, MUST DEFAULT')             LYR11800
 9002 FORMAT ('0 DATA USED TO CALC HM -'/' LEVEL  PRESS.   TEMP.    ELEVLYR11900
     1.  VIRT TEMP U-SPD COMP V-SPD COMP'/6X,'(PASCALS) (DEG K)    (M)  LYR12000
     2  (DEG K)     (M/S)       (M/S)')                                 LYR12100
 9003 FORMAT (2X,I2,F10.2,3F9.2,F10.2,F11.2)                            LYR12200
      END                                                               LYR12300
```

```
      SUBROUTINE LYRDHM(HMLYR,SFSZ)                                     LYD00100
      INCLUDE TIMES,LIST                                                LYD00200
C                                                                       LYD00300
C     SUBROUTINE PROVIDES DEFAULT MIXING LAYER DEPTH DEPENDING ON THE   LYD00400
C     HOUR AND MONTH                                                    LYD00500
C                                                                       LYD00600
      DIMENSION IDFHM(12,16)                                            LYD00700
      DATA IDFHM/ 3*30,50,11*30,107,110,139,103,35,6*30,114,326,467,403,LYD00800
     *329,238,129,89,2*30,64,122,387,740,865,691,688,615,490,271,165,96,LYD00900
     *232,329,832,1255,1298,1274,1034,1042,933,539,400,239,434,639,1247,LYD01000
     *1761,1907,2053,1342,1431,1343,873,593,369,645,1054,1518,2177,2521,LYD01100
     *2654,1868,1840,1708,1217,792,533,822,1363,1804,2513,2915,2983,    LYD01200
     *2388,2347,1896,1554,1025,734,907,1543,2145,2872,3195,3162,2503,   LYD01300
     *2749,1940,1711,1126,718,753,1586,2276,2843,3292,3144,2116,2484,   LYD01400
     *2038,1344,1039,435,386,1148,1835,1954,2727,3065,1279,1572,1762,   LYD01500
     *696,669,206,102,436,946,824,1578,2606,516,816,932,280,231,102,30, LYD01600
     *68,255,229,549,1466,191,398,252,181,71,35,30,111,30,63,77,427,172,LYD01700
     *115,30,164,45,2*30,142,4*30,153,2*30,77,14*30/                    LYD01800
C     DETERMINE HOUR INDEX                                              LYD01900
      IF (HOUR .GT. 4.AND.HOUR .LT. 14) GO TO 10                        LYD02000
      IHOUR = HOUR-13                                                   LYD02100
      IF (IHOUR .LE. 0) IHOUR = 24+IHOUR                                LYD02200
      GO TO 20                                                          LYD02300
   10 IHOUR = 16                                                        LYD02400
   20 CONTINUE                                                          LYD02500
      HMLYR = FLOAT(IDFHM(MONTH,IHOUR))+SFSZ                            LYD02600
      RETURN                                                            LYD02700
      END                                                               LYD02800
```

```
      SUBROUTINE CLSSTN(NSTATN,X,Y,IDXSTA,STADST,STNMAX,DISTMN)        CLS00100
      INCLUDE NEWPRM,LIST                                              CLS00200
C                                                                      CLS00300
C     DETERMINE THE CLOSEST STATION WITHIN DISTANCE DISTMN AND ORDER   CLS00400
C     STATIONS IN ASCENDING ORDER OF DISTANCE                          CLS00500
      DIMENSION X(1),Y(1),IDXSTA(1),STADST(1)                          CLS00600
      INTEGER STNMAX                                                   CLS00700
      NSTATN = 0                                                       CLS00800
      XSOURS = XSOURN*1.0E-3                                           CLS00900
      YSOURS = YSOURN*1.0E-3                                           CLS01000
      N = STNMAX                                                       CLS01100
      IF (N .GT. 25) N = 25                                            CLS01200
      DO 100 I=1,N                                                     CLS01300
C     CALCULATE DISTANCE FROM RELEASE TO STATION                       CLS01400
      D = SQRT((X(I)-XSOURS)**2+(Y(I)-YSOURS)**2)                      CLS01500
      IF (D .GT. DISTMN) GO TO 100                                     CLS01600
      NSTATN = NSTATN +1                                               CLS01700
      IDXSTA(NSTATN) = I                                               CLS01800
      STADST(NSTATN) = D                                               CLS01900
  100 CONTINUE                                                         CLS02000
      IF (NSTATN .LT. 2 ) GO TO 120                                    CLS02100
      DO 110 J=2,NSTATN                                                CLS02200
      DO 110 I=2,NSTATN                                                CLS02300
      IF (STADST(I-1) .LE. STADST(I)) GO TO 110                        CLS02400
      D = STADST(I)                                                    CLS02500
      K = IDXSTA(I)                                                    CLS02600
      STADST(I) = STADST(I-1)                                          CLS02700
      IDXSTA(I) = IDXSTA(I-1)                                          CLS02800
      STADST(I-1) = D                                                  CLS02900
      IDXSTA(I-1) = K                                                  CLS03000
  110 CONTINUE                                                         CLS03100
  120 RETURN                                                           CLS03200
      END                                                              CLS03300
```

```
      COMPILER(DIAG=3)                                                  UAD00100
      SUBROUTINE UADATA(STN,RUN,FLAG)                                   UAD00200
      INCLUDE ABUFER                                                    UAD00300
      INCLUDE OBUFER                                                    UAD00400
      INCLUDE FILES                                                     UAD00500
      INCLUDE TDATA                                                     UAD00600
      INCLUDE WDATA                                                     UAD00700
      INCLUDE TRPDTA                                                    UAD00800
      REAL      DP(40),D(40),S(40)                                      UAD00900
      REAL      ANGLE,OFFSET,PO,RADPDG                                  UAD01000
      REAL      GAMA,PSTD(2),TSTD(2),ZSTD                               UAD01100
      INTEGER   BLOCK,NEWBLK                                            UAD01200
      INTEGER   STN,RUN                                                 UAD01300
      INTEGER TADS,WADS                                                 UAD01400
      INTEGER   FLAG                                                    UAD01500
      EQUIVALENCE (DP,QS),(D,UW),(S,VW)                                 UAD01600
      DATA PO,RADPDG /100000.,0.01745329252/                            UAD01700
      DATA GAMA, PSTD,              TSTD,             ZSTD              UAD01800
     *   /.0065, 101325.,20980., 288.15,216.65, 11476./                 UAD01900
      DATA CONSTA /0.0341672473/                                        UAD02000
C  EXTRACT DATA                                                         UAD02100
      NEWBLK = TADRES(STN,RUN)/10000                                    UAD02200
      IF(NEWBLK.EQ.BLOCK) GO TO 50                                      UAD02300
      BLOCK = NEWBLK                                                    UAD02400
      CALL BLKIN(5000,A(1),BLOCK,UAFILE,ISTAT)                          UAD02500
   50 CONTINUE                                                          UAD02600
      NEWBLK = NEWBLK*10000                                             UAD02700
      TADS = TADRES(STN,RUN) + 1 - NEWBLK                               UAD02800
      WADS = WADRES(STN,RUN) + 1 - NEWBLK                               UAD02900
      FLAG = FLD(2,2,A(TADS-1))                                         UAD03000
      IF(FLD(0,6,A(WADS)).EQ.0) GO TO 60                                UAD03100
      D(1) = 0.0                                                        UAD03200
      S(1) = 0.0                                                        UAD03300
      ZW(1) = UASTNZ(STN)                                               UAD03400
      KMAX = KMAX +  1                                                  UAD03500
      GO TO 70                                                          UAD03600
   60 CONTINUE                                                          UAD03700
      FLAG = FLAG+4                                                     UAD03800
```

```
   70 CONTINUE                                                  UAD03900
      IF(IMAX.GT.40) IMAX = 40                                  UAD04000
      CALL XTRCUA(A(TADS),PS,IMAX,40,FLAG,1)                    UAD04100
      IF(IMAX.LE.1) GO TO 800                                   UAD04200
      IF(KMAX.GT.40) KMAX =  40                                 UAD04300
      CALL XTRCUA(A(WADS),UW,KMAX,40,FLAG,2)                    UAD04400
      IF(KMAX.LE.1) GO TO 800                                   UAD04500
      NMAX = FLD(18,9,A(TADS-1)) / 3                            UAD04600
      IF(NMAX.EQ.0) GO TO 80                                    UAD04700
      IF(NMAX.GT.3) NMAX = 3                                    UAD04800
      TADS = TADS + FLD(9,9,A(TADS-1)) - 1                      UAD04900
      CALL XTRCUA(A(TADS),PT,NMAX,3,FLAG,3)                     UAD05000
   80 CONTINUE                                                  UAD05100
C     EVALUATE WIND COMPONENTS RELATIVE TO BNL UTM GRID         UAD05200
      OFFSET = UASTNA(STN)                                      UAD05300
      DO 100 K = 1,KMAX                                         UAD05400
      ANGLE = RADPDG*(D(K) + OFFSET)                            UAD05500
      UW(K) = -S(K)*SIN(ANGLE)                                  UAD05600
      VW(K) =-S(K)*COS(ANGLE)                                   UAD05700
  100 CONTINUE                                                  UAD05800
C  EVALUATE PRESSURE LEVEL DATA                                 UAD05900
      IF(FLD(32,1,FLAG).NE.0) GO TO 135                         UAD06000
      K = 2                                                     UAD06100
      ZS(1) = UASTNZ(STN)                                       UAD06200
  105 CONTINUE                                                  UAD06300
      IF(DP(K).LT.0) GO TO 110                                  UAD06400
      DZ = 62.2*SATVP(TS(K) - DP(K)) / PS(K)                    UAD06500
      DZ = DZ / (1.-0.61*DZ)                                    UAD06600
      TV(K) = TS(K) * (1.0 + 0.61 * DZ)                         UAD06700
      GO TO 115                                                 UAD06800
  110 CONTINUE                                                  UAD06900
      TV(K) = TS(K)                                             UAD07000
  115 CONTINUE                                                  UAD07100
      IF(ZS(K).GT.0.) GO TO 120                                 UAD07200
      K = K + 1                                                 UAD07300
      GO TO 105                                                 UAD07400
  120 CONTINUE                                                  UAD07500
      IF(K.EQ.2) GO TO 130                                      UAD07600
```

```
      DO 125 I = K,3,-1                                                   UAD07700
      DELTA(I) = ALOG(PS(I) / PS(I-1)) / ALOG(TV(I) / TV(I-1))            UAD07800
      ZS(I-1) = ZS(I) + DELTA(I) * (TV(I) - TV(I-1)) / CONSTA             UAD07900
  125 CONTINUE                                                            UAD08000
  130 CONTINUE                                                            UAD08100
      DZ = ZS(2) - ZS(1)                                                  UAD08200
      TS(1) = TS(2) + GAMA * DZ                                           UAD08300
      DP(1) = DP(2)                                                       UAD08400
      PS(1) = PS(2) * (TS(1) / TS(2))**(CONSTA / GAMA)                    UAD08500
  135 CONTINUE                                                            UAD08600
      K = 2                                                               UAD08700
      DZW = (ZW(2) - ZW(1))                                               UAD08800
      DO 400 I = 1,IMAX                                                   UAD08900
      IF(DP(I).LT.0.) GO TO 140                                           UAD09000
      QS(I) = 62.2*SATVP(TS(I) - DP(I))/PS(I)                             UAD09100
      QS(I) = QS(I)/(1. - 0.61*QS(I))                                     UAD09200
      TV(I) = (1. + 0.61*QS(I))*TS(I)                                     UAD09300
      GO TO 160                                                           UAD09400
  140 CONTINUE                                                            UAD09500
      QS(I) = 0.0                                                         UAD09600
      TV(1) = TS(I)                                                       UAD09700
  160 CONTINUE                                                            UAD09800
      TA(1) = TS(I)*(PO/PS(I))**0.286                                     UAD09900
      IF(I.EQ.1) GO TO 300                                                UAD10000
      IF(TV(1).EQ.TV(I-1)) GO TO 200                                      UAD10100
      DELTA(I) = ALOG(PS(I)/PS(I-1))/ALOG(TV(I)/TV(I-1))                  UAD10200
      IF (ZS(I).LT.0.)                                                    UAD10300
     1ZS(I) = ZS(I-1) - DELTA(I)*(TV(I) - TV(I-1))/CONSTA                 UAD10400
      GO TO 300                                                           UAD10500
  200 CONTINUE                                                            UAD10600
      DELTA(I) = 0.0                                                      UAD10700
      IF (ZS(I).LT.0.)                                                    UAD10800
     1ZS(I) = ZS(I-1) - TV(I)*ALOG(PS(I)/PS(I-1))/CONSTA                  UAD10900
  300 CONTINUE                                                            UAD11000
      DZ = ZW(K) - ZS(I)                                                  UAD11100
      IF(DZ) 320,340,360                                                  UAD11200
  320 CONTINUE                                                            UAD11300
      IF(K.GE.KMAX) GO TO 500                                             UAD11400
```

```
      K = K+1                                                           UAD11500
      DZW = ZW(K) - ZW(K-1)                                             UAD11600
      GO TO 500                                                         UAD11700
  340 CONTINUE                                                          UAD11800
      US(I) = UW(K)                                                     UAD11900
      VS(I) = VW(K)                                                     UAD12000
      GO TO 370                                                         UAD12100
  360 CONTINUE                                                          UAD12200
      DZ = DZ/DZW                                                       UAD12300
      US(I) = UW(K) - DZ*(UW(K) - UW(K-1))                              UAD12400
      VS(I) = VW(K) - DZ*(VW(K) - VW(K-1))                              UAD12500
  370 CONTINUE                                                          UAD12600
      IF(PS(I).LT.PSTD(2)) GO TO 380                                    UAD12700
      PD(I) = ZS(I) - TSTD(1)*(1.0-(PS(I)/PSTD(1))**(GAMA/CONSTA))/GAMA UAD12800
      GO TO 400                                                         UAD12900
  380 CONTINUE                                                          UAD13000
      PD(I) = ZS(I) - ZSTD + TSTD(2)*ALOG(PS(I)/PSTD(2))/CONSTA         UAD13100
  400 CONTINUE                                                          UAD13200
      GO TO 600                                                         UAD13300
  500 CONTINUE                                                          UAD13400
      IMAX = I-1                                                        UAD13500
  600 CONTINUE                                                          UAD13600
      IF(NMAX.EQ.0) GO TO 760                                           UAD13700
      I = 1                                                             UAD13800
      K = 1                                                             UAD13900
  610 CONTINUE                                                          UAD14000
      K = K + 1                                                         UAD14100
      IF(K.GT.IMAX) GO TO 720                                           UAD14200
      IF(PS(K) - PT(I)) 640,620,610                                     UAD14300
  620 CONTINUE                                                          UAD14400
      QT(I) = QS(K)                                                     UAD14500
      TB(I) =TV(K)                                                      UAD14600
      TI(I) = TA(K)                                                     UAD14700
      TD(I) = PD(K)                                                     UAD14800
      ZT(I) = ZS(K)                                                     UAD14900
      IF((UT(I).LT.0.0).OR.(VT(I).LT.0.0)) GO TO 630                    UAD15000
      ANGLE = RADPDG*(UT(I) + OFFSET)                                   UAD15100
      UT(I) = -VT(I) * SIN(ANGLE)                                       UAD15200
```

```
      VT(I) = -VT(I) * COS(ANGLE)                                       UAD15300
      US(K) = UT(I)                                                     UAD15400
      VS(K) = VT(I)                                                     UAD15500
      GO TO 700                                                         UAD15600
  630 CONTINUE                                                          UAD15700
      UT(I) = US(K)                                                     UAD15800
      VT(I) = VT(K)                                                     UAD15900
      GO TO 700                                                         UAD16000
  640 CONTINUE                                                          UAD16100
      IF(QT(I).LT.0.) GO TO 650                                         UAD16200
      QT(I) = 62.2 * SATVP(TT(I) - QT(I)) / PT(I)                       UAD16300
      QT(I) = QT(I) / (1.0-0.61*QT(I))                                  UAD16400
      TB(I) = (1.0+0.61*QT(I)) * TT(I)                                  UAD16500
      GO TO 660                                                         UAD16600
  650 CONTINUE                                                          UAD16700
      QT(I) = 0.0                                                       UAD16800
      TB(I) = TT(I)                                                     UAD16900
  660 CONTINUE                                                          UAD17000
      TI(I) = TT(I) * (PO/PT(I)) ** 0.286                               UAD17100
      IF(TB(I).EQ.TV(K-1)) GO TO 670                                    UAD17200
      DZW = ALOG(PT(I)/PS(K-1)) / ALOG(TB(I) / TV(K-1))                 UAD17300
      ZT(I) = ZS(K-1) - DZW * (TB(I) - TV(K-1)) / CONSTA                UAD17400
      GO TO 680                                                         UAD17500
  670 CONTINUE                                                          UAD17600
      ZT(I) = ZS(I-1) - TB(I) * ALOG(PT(I) / PS(K-1)) / CONSTA          UAD17700
  680 CONTINUE                                                          UAD17800
      IF((UT(I).LT.0.0).OR.(VT(I).LT.0.0)) GO TO 690                    UAD17900
      ANGLE = RADPDG * (UT(I) + OFFSET)                                 UAD18000
      UT(I) = -VT(I) * SIN(ANGLE)                                       UAD18100
      VT(I) = -VT(I) * COS(ANGLE)                                       UAD18200
      GO TO 700                                                         UAD18300
  690 CONTINUE                                                          UAD18400
      DZW = (ZT(I) - ZS(K-1)) / (ZS(K) - ZS(K-1))                       UAD18500
      UT(I) = US(K-1) + (US(K) - US(K-1)) * DZW                         UAD18600
      VT(I) = VS(K-1) + (VS(K) - VS(K-1)) * DZW                         UAD18700
  700 CONTINUE                                                          UAD18800
      I = I + 1                                                         UAD18900
      IF(I.LE.NMAX) GO TO 610                                           UAD19000
```

```
      GO TO 760                                                   UAD19100
  720 CONTINUE                                                    UAD19200
      NMAX = I - 1                                                UAD19300
  760 CONTINUE                                                    UAD19400
      RETURN                                                      UAD19500
  800 CONTINUE                                                    UAD19600
C  INADEQUATE DATA, ELIMINATE OB FROM INVENTORY                   UAD19700
      WRITE(6,900) FLAG,IMAX,KMAX,TADS,WADS,                      UAD19800
     *               A(TADS-1),A(TADS),A(WADS-1),A(WADS)          UAD19900
  900 FORMAT(' FL,IMX,KMX,TAD,WAD',5I10/                          UAD20000
     *         ' A(T-1),A(T),A(W-1),A(W)',4(2XO12))               UAD20100
      FLAG = 0                                                    UAD20200
      TLVLMX(STN,RUN) = 0                                         UAD20300
      WLVLMX(STN,RUN) = 0                                         UAD20400
      RETURN                                                      UAD20500
      END                                                         UAD20600
```

```
      SUBROUTINE XTRCUA (A,B,LEVEL,N,FLAG,BRANCH)                       XTR00100
      REAL      B(N,4)                                                  XTR00200
      INTEGER   A(1),BRANCH,FLAG,INDI,LEVEL,SIGN(3)                     XTR00300
      DATA SIGN /0000000007777,                                         XTR00400
     *           0000000001777,                                         XTR00500
     *           0000000377777 /                                        XTR00600
C                                                                       XTR00700
      J = 1                                                             XTR00800
      L = 1                                                             XTR00900
      IF(BRANCH - 1) 1000,20,400                                        XTR01000
   20 CONTINUE                                                          XTR01100
      INDI = FLD(0,6,A(J))                                              XTR01200
      IF(INDI.EQ.0) GO TO 80                                            XTR01300
      INDI = FLD(13,12,A(J))                                            XTR01400
      J = J + 2                                                         XTR01500
      IF(INDI.NE.SIGN(1)) GO TO 60                                      XTR01600
      LEVEL = LEVEL - 1                                                 XTR01700
      IF(LEVEL.GT.0) GO TO 20                                           XTR01800
      GO TO 700                                                         XTR01900
   60 CONTINUE                                                          XTR02000
      L = 2                                                             XTR02100
      LEVEL = LEVEL + 1                                                 XTR02200
      IF(LEVEL.GT.N) LEVEL = N                                          XTR02300
      GO TO 100                                                         XTR02400
   80 CONTINUE                                                          XTR02500
      FLAG = FLAG + 8                                                   XTR02600
  100 CONTINUE                                                          XTR02700
      INDI = FLD(0,6,A(J))                                              XTR02800
      IF(INDI.GT.2) GO TO 360                                           XTR02900
      INDI = FLD(13,12,A(J))                                            XTR03000
      IF(INDI.EQ.SIGN(1)) GO TO 360                                     XTR03100
      B(L,3) = FLOAT(INDI)/10.                                          XTR03200
      INDI = FLD(26,10,A(J))                                            XTR03300
      IF(INDI.NE.SIGN(2)) GO TO 160                                     XTR03400
      B(L,2) = -1.0                                                     XTR03500
      GO TO 200                                                         XTR03600
  160 CONTINUE                                                          XTR03700
      B(L,2) = FLOAT(INDI)/10.0                                         XTR03800
```

```
      200 CONTINUE                                                      XTR03900
          INDI = FLD(1,17,A(J+1))                                       XTR04000
          IF(INDI.EQ.SIGN(3)) GO TO 360                                 XTR04100
          B(L,1) = INDI*10                                              XTR04200
          INDI = FLD(19,17,A(J+1))                                      XTR04300
          IF(INDI.NE.SIGN(3)) GO TO 260                                 XTR04400
          B(L,4) = -1.0                                                 XTR04500
          GO TO 300                                                     XTR04600
      260 CONTINUE                                                      XTR04700
          B(L,4) = INDI                                                 XTR04800
      300 CONTINUE                                                      XTR04900
          L = L+1                                                       XTR05000
      320 CONTINUE                                                      XTR05100
          IF(L.GT.LEVEL) GO TO 700                                      XTR05200
          J = J+2                                                       XTR05300
          GO TO 100                                                     XTR05400
      360 CONTINUE                                                      XTR05500
          LEVEL = LEVEL - 1                                             XTR05600
          GO TO 320                                                     XTR05700
      400 CONTINUE                                                      XTR05800
          IF(BRANCH.NE.2) GO TO 800                                     XTR05900
          IF(FLD(2,1,FLAG).EQ.1) GO TO 420                              XTR06000
          L = 2                                                         XTR06100
      420 CONTINUE                                                      XTR06200
          INDI = FLD(0,6,A(J))                                          XTR06300
          IF(INDI.GT.2) GO TO 660                                       XTR06400
          INDI = FLD(12,12,A(J))                                        XTR06500
          IF(INDI.EQ.SIGN(1)) GO TO 660                                 XTR06600
          B(L,1) = INDI                                                 XTR06700
          INDI = FLD(24,12,A(J))                                        XTR06800
          IF(INDI.EQ.SIGN(1)) GO TO 660                                 XTR06900
          B(L,2) = FLOAT(INDI)/10.                                      XTR07000
      500 CONTINUE                                                      XTR07100
          INDI = FLD(19,17,A(J+1))                                      XTR07200
          IF(INDI.EQ.SIGN(3)) GO TO 660                                 XTR07300
          B(L,3) = INDI                                                 XTR07400
      600 CONTINUE                                                      XTR07500
          L = L+1                                                       XTR07600
```

```
  620 CONTINUE                                                XTR07700
      IF(L.GT.LEVEL) GO TO 700                                XTR07800
      J = J+2                                                 XTR07900
      GO TO 420                                               XTR08000
  660 CONTINUE                                                XTR08100
      LEVEL = LEVEL - 1                                       XTR08200
      GO TO 620                                               XTR08300
  700 CONTINUE                                                XTR08400
      IF(LEVEL.LT.0) LEVEL = 0                                XTR08500
      RETURN                                                  XTR08600
  800 CONTINUE                                                XTR08700
      IF(BRANCH.NE.3) GO TO 1000                              XTR08800
      INDI = FLD(0,6,A(J))                                    XTR08900
      IF(INDI.NE.5) GO TO 960                                 XTR09000
      INDI = FLD(13,12,A(J))                                  XTR09100
      IF(INDI.EQ.SIGN(1)) GO TO 980                           XTR09200
      B(L,3) = FLOAT(INDI)/10.                                XTR09300
      INDI = FLD(26,10,A(J))                                  XTR09400
      IF(INDI.NE.SIGN(2)) GO TO 820                           XTR09500
      B(L,2) = -1.0                                           XTR09600
      GO TO 840                                               XTR09700
  820 CONTINUE                                                XTR09800
      B(L,2) = FLOAT(INDI)/10.0                               XTR09900
  840 CONTINUE                                                XTR10000
      INDI = FLD(1,17,A(J+1))                                 XTR10100
      IF(INDI.EQ.SIGN(3)) GO TO 980                           XTR10200
      B(L,1) = INDI*10                                        XTR10300
      INDI = FLD(19,17,A(J+1))                                XTR10400
      IF(INDI.NE.SIGN(3)) GO TO 860                           XTR10500
      B(L,4) = -1.0                                           XTR10600
      GO TO 900                                               XTR10700
  860 CONTINUE                                                XTR10800
      B(L,4) = INDI                                           XTR10900
  900 CONTINUE                                                XTR11000
      INDI = FLD(12,12,A(J+2))                                XTR11100
      IF(INDI.EQ.SIGN(1)) GO TO 960                           XTR11200
      B(L,7) = INDI                                           XTR11300
      INDI = FLD(24,12,A(J+2))                                XTR11400
```

```
      IF(INDI.EQ.SIGN(1)) GO TO 960                                     XTR11500
      B(L,8) = FLOAT(INDI)/10.                                          XTR11600
  920 CONTINUE                                                          XTR11700
      L = L + 1                                                         XTR11800
  940 CONTINUE                                                          XTR11900
      IF(L.GT.LEVEL) GO TO 700                                          XTR12000
      J = J + 3                                                         XTR12100
      GO TO 800                                                         XTR12200
  960 CONTINUE                                                          XTR12300
      B(L,7) = -1.0                                                     XTR12400
      B(L,8) = -1.0                                                     XTR12500
      GO TO 920                                                         XTR12600
  980 CONTINUE                                                          XTR12700
      LEVEL = LEVEL - 1                                                 XTR12800
      GO TO 920                                                         XTR12900
 1000 CONTINUE                                                          XTR13000
      STOP                                                              XTR13100
      END                                                               XTR13200
```

```
      FUNCTION SATVP(T)                                                 SAT00100
C                                                                       SAT00200
      SATVP = 10.0**(23.83224-2949.076/T-5.02808*ALOG10(T)              SAT00300
     1 -1.3816E-7*10.**(11.344-.0303998*T)+8.1328E-3*10.**(3.49149-     SAT00400
     2 1302.8844/T))                                                    SAT00500
      RETURN                                                            SAT00600
      END                                                               SAT00700
```

```
      COMPILER(DIAG=3)                                                   SFC00100
      SUBROUTINE SFCDTA(STN,RUN,FLAG)                                    SFC00200
C   THE PURPOSE OF THIS ROUTINE IS TO EXTRACT SURFACE OBSERVATION DATA  SFC00300
C     FROM THE DATA BASE .                                               SFC00400
      INCLUDE SFDATA,LIST                                                SFC00500
      INCLUDE ABUFER,LIST                                                SFC00600
      INCLUDE SBUFER,LIST                                                SFC00700
      DIMENSION WW(4)                                                    SFC00800
      EQUIVALENCE (WW,WW1),(TS,TSS)                                      SFC00900
      INTEGER WW                                                         SFC01000
      INTEGER JUNK/-1/,RUN,FLAG,STN                                      SFC01100
      II = STN                                                           SFC01200
C  FIND THE ADDRESS FOR THE DATA CORRESPONDING TO THE GIVEN STATION AND SFC01300
C   TIME . IF THIS ADDRESS HAS NOT BEEN SET (IT IS ZERO) , RETURN TO    SFC01400
C   THE CALLING PROGRAM . OTHERWISE , FIND J , WHERE THE FIRST WORD OF  SFC01500
C   THE 11-WORD LOGICAL RECORD IS THE J-TH ELEMENT IN ARRAY A .         SFC01600
      FLAG = 0                                                           SFC01700
      IF (SADRES(II,RUN).EQ.0)  RETURN                                   SFC01800
      I = SADRES(II,RUN)/10000                                           SFC01900
      J = SADRES(II,RUN) - I*10000                                       SFC02000
      FLAG = FLD(0,18,A(J))                                              SFC02100
C    RETURN TO THE CALLING PROGRAM IF THE VALUE OF FLAG IS ZERO .        SFC02200
      IF (FLAG.EQ.0)   RETURN                                            SFC02300
C  EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 1 .                  SFC02400
      CALL TESTFD(0,12,A(J+2),INDI)                                      SFC02500
      WD = INDI                                                          SFC02600
      CALL TESTFD(12,12,A(J+2),INDI)                                     SFC02700
      IF (INDI.EQ.JUNK) INDI = -10                                       SFC02800
      WS = FLOAT(INDI)/10.                                               SFC02900
      CALL TESTFD(24,12,A(J+2),INDI)                                     SFC03000
      IF (INDI.EQ.0) INDI = -10                                          SFC03100
      WG = FLOAT(INDI)/10.                                               SFC03200
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 2 .                 SFC03300
      CALL TESTFD(0,18,A(J+3),INDI)                                      SFC03400
      IF (INDI.EQ.JUNK) INDI = -10                                       SFC03500
      SLP = FLOAT(INDI)/10.                                              SFC03600
      CALL TESTFD(18,18,A(J+3),BT)                                       SFC03700
      IF (BT.EQ.0) BT = JUNK                                             SFC03800
```

```
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 3 .                    SFC03900
       CALL TESTFD(0,18,A(J+4),INDI)                                         SFC04000
       IF (INDI.EQ.JUNK) INDI = -10                                          SFC04100
       TS = FLOAT(INDI)/10.                                                  SFC04200
       CALL TESTFD(18,18,A(J+4),INDI)                                        SFC04300
       IF (INDI.EQ. JUNK) INDI = -10                                         SFC04400
       TD = FLOAT(INDI)/10.                                                  SFC04500
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 4 .                    SFC04600
       CALL TESTFD(0,18,A(J+5),INDI)                                         SFC04700
       IF (INDI.EQ.JUNK) INDI = -100                                         SFC04800
       AS = FLOAT(INDI)/100.                                                 SFC04900
       CALL TESTFD(18,18,A(J+5),INDI)                                        SFC05000
       PA = INDI                                                             SFC05100
C    EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 5 .                   SFC05200
       CALL TESTFD(0,6,A(J+6),TSC)                                           SFC05300
       CALL TESTFD(6,6,A(J+6),PW)                                            SFC05400
       CALL TESTFD(18,18,A(J+6),VV)                                          SFC05500
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 6 . TEST THAT THERE SFC05600
C     ARE CODED DATA ON THE PRESENT WEATHER .                               SFC05700
       IF (A(J+7).NE.0) GO TO 15                                             SFC05800
       DO 20 I = 1,4                                                         SFC05900
    20 WW(I) = JUNK                                                          SFC06000
           GO TO 16                                                          SFC06100
    15 DO 25 I = 1,4                                                         SFC06200
       CALL TESTFD((I-1)*9,9,A(J+7),WW(I))                                   SFC06300
    25 CONTINUE                                                              SFC06400
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 8 .                    SFC06500
    16 CALL TESTFD(6,6,A(J+9),NH)                                            SFC06600
       CALL TESTFD(12,6,A(J+9),CL)                                           SFC06700
       CALL TESTFD(18,6,A(J+9),H)                                            SFC06800
       CALL TESTFD(24,6,A(J+9),CM)                                           SFC06900
       CALL TESTFD(30,6,A(J+9),CH)                                           SFC07000
C   EXTRACT THE DATA IN THE (SURFACE) MANDATORY WORD 9 .                    SFC07100
       CALL TESTFD(6,6,A(J+10),NS)                                           SFC07200
       CALL TESTFD(12,6,A(J+10),K)                                           SFC07300
       CALL TESTFD(18,6,A(J+10),CT)                                          SFC07400
       CALL TESTFD(24,12,A(J+10),HSHS)                                       SFC07500
       RETURN                                                                SFC07600
```

```
77*        END                                                    SFC07700
```

```
      COMPILER(DIAG=3)                                                  TES00100
      SUBROUTINE TESTFD(LFTBIT,LGTBIT,WORD,IVALUE)                      TES00200
C  LFTBIT IS THE POSITION OF THE LEFT END OF THE FIELD BEING EXTRACTED ,TES00300
C     COUNTED FROM 0 (MOST SIGNIFICANT BIT) TO 35 (LEAST SIGNIFICANT    TES00400
C     BIT) .                                                            TES00500
C  LGTBIT IS THE LENGTH IN BITS OF THE FIELD BEING EXTRACTED .          TES00600
C  WORD IS THE WORD FROM WHICH THE FIELD IS BEING EXTRACTED .           TES00700
C  IVALUE IS THE INTEGER VALUE OF THE FIELD BEING EXTRACTED . IT MAY    TES00800
C   NEED TO BE CONVERTED TO REAL AND/OR SCALED BY A CONSTANT FACTOR .   TES00900
      IVALUE = FLD(IABS(LFTBIT),LGTBIT,WORD)                            TES01000
C  TEST WHETHER THE FIELD EXTRACTED LACKS MEANINGFUL DATA . IF IT DOES ,TES01100
C                       RETURN TO CALLING PROGRAM .                     TES01200
      NONSNS = 2**LGTBIT - 1                                            TES01300
      IF (IVALUE.NE.NONSNS)  GO TO 11                                   TES01400
      IVALUE = -1                                                       TES01500
      RETURN                                                            TES01600
C   TEST WHETHER THE FIELD EXTRACTED IS A NEGATIVE NUMBER . IF IT IS ,  TES01700
C      CONVERT IT BY SUBTRACTING OUT A PROPER CONSTANT .                TES01800
   11 IBIT = FLD(IABS(LFTBIT),1,WORD)                                   TES01900
      IF (IBIT.EQ.1)  IVALUE = IVALUE - NONSNS                          TES02000
      RETURN                                                            TES02100
      END                                                               TES02200
```

```
      COMPILER (DIAG=3)
      SUBROUTINE GWCDTA(GRID,RUN,FLAG)
      INCLUDE ABUFER,LIST
      INCLUDE FILES
      INCLUDE GBUFER
      INCLUDE GWDATA
      INTEGER FLAG,GRID,RUN,ZS(4)
      INTEGER A,B(4,6),X,Y
      INTEGER ADRES,BLOCK
      INTEGER OCT/0777777000000/
      DATA P /100000.,85000.,70000.,50000./
      DATA ZS/110,1460,3010,5570/
      IF(RUN.EQ.BLOCK) GO TO 50
      BLOCK = RUN
      CALL BLKIN(5700,A(1),BLOCK,GWCFIL,I)
   50 CONTINUE
      ADRES = GTIME - ANLTIM(RUN)
      IF(ADRES.LT.0) GO TO 600
      IF(ADRES.GT.24) ADRES = ADRES - 76
      IF(ADRES.GT.18) GO TO 800
      ADRES = 300*ADRES + 25*(GRID-1)
      DO 150 J = 1,6
      DO 100 K = 1,2
      L = K+K
      M = 2*(J-1) + ADRES
      X = FLD(18,18,A(M+K))
      Y = FLD(0,18,A(M+K))
      IF(X.GT.32768) X = OR(X,OCT)
      IF(Y.GT.32768) Y = OR(Y,OCT)
      B(L-1,J) = X
      B(L,J) = Y
  100 CONTINUE
  150 CONTINUE
      DO 200 L = 1,4
      U(L) = B(L,1)
      V(L) = B(L,2)
      Z(L) = B(L,4) + ZS(L)
      T(L) = B(L,5)
```

AD-A038 399 CRAMER (H E) CO INC SALT LAKE CITY UTAH F/G 4/2
MIXING-LAYER ANALYSIS ROUTINE AND TRANSPORT/DIFFUSION APPLICATI--ETC(U)
MAR 77 R K DUMBAULD, J R BJORKLUND DAAD07-76-C-0023
UNCLASSIFIED TR-76-106-01 ECOM-77-2 NL

3 OF 3
AD
A038399

END
DATE
FILMED
5-77

```
      Q(L) = B(L,6)                                                     GWC03900
      Q(L) = T(L)-Q(L)                                                  GWC04000
      IF (Q(L) .LT. 0.0) Q(L) = 0.0                                     GWC04100
      IF (Q(L) .GT. 0.0.AND.P(L) .GT. 0.0) Q(L)=62.2*SATVP(Q(L))/P(L)   GWC04200
      Q(L) = Q(L)/(1. - 0.61*Q(L))                                      GWC04300
  200 CONTINUE                                                          GWC04400
      FLAG = 1                                                          GWC04500
      IF(RUN.EQ.41) FLAG = 0                                            GWC04600
      RETURN                                                            GWC04700
  800 CONTINUE                                                          GWC04800
      WRITE(6,900)                                                      GWC04900
      FLAG = 0                                                          GWC05000
      RETURN                                                            GWC05100
  900 FORMAT( 'IMPROPER GWC TIME OR BLOCK')                             GWC05200
      END                                                               GWC05300
```

```
      SUBROUTINE DAYNIT(LON,LAT,SNRISE,SUNSET,ALPHA)                    DAY00100
C     THIS SUBROUTINE CALCULATES THE TIME OF 1 HOUR BEFORE SUNSET AND   DAY00200
C     1 HOUR AFTER SUNRISE AND THE SOLAR ALTITUDE ALPHA                 DAY00300
      REAL LON,LAT                                                      DAY00400
      INCLUDE TIMES,LIST                                                DAY00500
      TIME = FLOAT(HOUR)+FLOAT(MINUTE)*1.6666666E-2                     DAY00600
C     LON IS POS. WEST OF GREENWICH UK.                                 DAY00700
      CPHI = LAT*.01745329                                              DAY00800
      SPHI = SIN(CPHI)                                                  DAY00900
      CPHI = COS(CPHI)                                                  DAY01000
      D1= (JULIAN+DAY-1.0)*360.0/365.242                                DAY01100
      D2 = D1*.01745329                                                 DAY01200
      D3 = 2*D2                                                         DAY01300
      CD1 = COS(D2)                                                     DAY01400
      SD1 = SIN(D2)                                                     DAY01500
      S2D1 = SIN(D3)                                                    DAY01600
      C2D1 = COS(D3)                                                    DAY01700
      GAM = 279.9348+D1+1.914827*SD1-7.9525E-2*CD1+1.9938E-2*S2D1-      DAY01800
     11.62E-3*C2D1                                                      DAY01900
      SD = .39785*SIN(GAM*.01745329)                                    DAY02000
      CD= SQRT(1.0-SD*SD)                                               DAY02100
      TM = 12.0+.12357*SD1-4.289E-3*CD1+.153809*S2D1+6.0783E-2*C2D1     DAY02200
      TH = ACOS((-1.5707E-2-SPHI*SD)/(CPHI*CD))*57.2957795              DAY02300
      SNRISE = (LON-TH)*6.6666666E-2+TM+1.0                             DAY02400
      SUNSET = (LON+TH)*6.6666666E-2+TM-1.0                             DAY02500
      ALPHA = SPHI*SD+CPHI*CD*COS(.01745329*(15.0*(TIME-TM)-LON))       DAY02600
      IF (ALPHA .GT. 1.0) ALPHA = 1.0                                   DAY02700
      IF (ALPHA .LT.-1.0) ALPHA =-1.0                                   DAY02800
      ALPHA = ASIN(ALPHA)*57.2957795                                    DAY02900
      IF (SNRISE .LT. 0.0) SNRISE = SNRISE+24.0                         DAY03000
      IF (SNRISE .GE. 24.0) SNRISE = SNRISE-24.0                        DAY03100
      IF (SUNSET .LT. 0.0) SUNSET = SUNSET+24.0                         DAY03200
      IF (SUNSET .GE. 24.0) SUNSET = SUNSET-24.0                        DAY03300
      RETURN                                                            DAY03400
      END                                                               DAY03500
```

```
      SUBROUTINE LONLAT(X,Y,LON,LAT)                                    LON00100
C     THIS ROUTINE CALCULATES THE LONGITUDE AND LATITUDE OF THE COORD   LON00200
C **  POSITION X,Y RELATIVE TO UTM ZONE - ZONENO = 13          **       LON00300
      REAL LON,LAT                                                      LON00400
      DOUBLE PRECISION ESQ,PI,MTOF,MTOFI,SECRD,FE,CONS1 , CONS2,        LON00500
     1CONS3, CONS4,A,SCALE,PPRD,SEPQ,SNLT,CSLT,CSSQ,EPSQ,SNONS,         LON00600
     2PHRD,PHIS,Q,SNLAT,CSLAT,SNSQ,CCSQ,TNLAT,TNSQ,ENU,ENSNS,QSQ,QFR,   LON00700
     3QSX,EPCSQ,SVN,EGHT,D6,DLAM,EPCS,QCU,QFV,SCLAT,ENSN5,ANINE,        LON00800
     4TEN,E5                                                            LON00900
      DATA ESQ/.006768658D0/,PI/3.1415926535898D0/,A/6378206.4D0/,      LON01000
     1ZONENO/13./,SCALE/.9996D0/,ISW/0/,CONS1/.24682D0/,CONS2/30.02335D0LON01100
     2/,CONS3/5078.64977D0/,CONS4/.157049981D0/,SEPQ/.0408887094D0/     LON01200
      CM = (6.0*ZONENO-183.0)*3600.                                     LON01300
      MTOF = 39.37/12.0                                                 LON01400
      MTOFI = 12.0/39.37                                                LON01500
      SECRD = 648000.0/PI                                               LON01600
      FE = 500000.0*MTOF                                                LON01700
      EPSQ = ESQ/(1.0-ESQ)                                              LON01800
      SNONS = 1.0/SECRD                                                 LON01900
      LAT = X*MTOF                                                      LON02000
      LON = Y*MTOF                                                      LON02100
      PPRD = (Y*CONS4*1.0E-6)/SCALE                                     LON02200
      SNLT = DSIN(PPRD)                                                 LON02300
      CSLT = DCOS(PPRD)                                                 LON02400
      CSSQ = CSLT*CSLT                                                  LON02500
      PHRD = ((CONS1*CSSQ+CONS2)*CSSQ+CONS3)*(SNLT*CSLT)*1.0E-6+PPRD    LON02600
      PHIS = PHRD*SECRD                                                 LON02700
      Q = ((LAT-FE)*MTOFI)*1.0E-6                                       LON02800
      IF (Q) 300,200,300                                                LON02900
  200 DLAM = 0.0                                                        LON03000
      LAT = PHIS/3600.0                                                 LON03100
      GO TO 400                                                         LON03200
  300 SNLAT = DSIN(PHRD)                                                LON03300
      CSLAT = DCOS(PHRD)                                                LON03400
      SNSQ = SNLAT*SNLAT                                                LON03500
      CCSQ = CSLAT*CSLAT                                                LON03600
      TNLAT = SNLAT/CSLAT                                               LON03700
      TNSQ = TNLAT*TNLAT                                                LON03800
```

```
      ENU = A/DSQRT(1.0-ESQ*SNSQ)                                       LON03900
      ENSNS = ENU*SNONS                                                 LON04000
      EPCS = EPSQ*CCSQ                                                  LON04100
      EPCSQ = 1.0+EPCS                                                  LON04200
      QSQ = Q*Q                                                         LON04300
      QCU = QSQ*Q                                                       LON04400
      QFR = QCU*Q                                                       LON04500
      QFV = QFR*Q                                                       LON04600
      QSX = QFV*Q                                                       LON04700
      SCLAT = 1.0/CSLAT                                                 LON04800
      ENSN5 = ENU**4*ENSNS                                              LON04900
      SVN = (((TNLAT/(2.0*ENU*ENSNS))*EPCSQ)/SCALE**2)*10.0D11          LON05000
      EGHT = ((TNLAT/(24.0*ENU**3*ENSNS))*(5.+3.*TNSQ+SEPG*(CCSQ-SNSQ)  LON05100
     1-(3.0*EPSQ**2*CCSQ)*(CCSQ+3.*SNSQ))/SCALE**4)*10.D23              LON05200
      D6 = (QSX*(TNLAT/(720.*ENU**5*ENSNS))*(61.+(45.*TNSQ)*(2.+TNSQ-   LON05300
     1EPSQ*SNSQ)+EPSQ*(107.*CCSQ-162.*SNSQ))/SCALE**6)*10.0D35          LON05400
      ANINE = (SCLAT/ENSNS)/SCALE * 1.0E6                               LON05500
      TEN = (SCLAT/(6.*ENU**2*ENSNS))*(1.+2.*TNSQ+EPCS)/SCALE**3*1.0D18 LON05600
      E5 = QFV*(SCLAT/(120.*ENSN5))*(5.+(4.*TNSQ)*(7.+6.*TNSQ)          LON05700
     1+(2.*EPSQ)*(3.*CCSQ+4.*SNSQ))/SCALE**5*1.0D30                     LON05800
      LAT = (PHIS-SVN*QSQ+EGHT*QFR-D6)/3600.0                           LON05900
      DLAM = ANINE*Q-TEN*QCU+E5                                         LON06000
  400 CONTINUE                                                          LON06100
      LON = -(CM+DLAM)/3600.0                                           LON06200
      RETURN                                                            LON06300
      END                                                               LON06400
```

```
      SUBROUTINE SOLVMX                                                 SOL00100
C     2-LAYER MESOSCALE MODEL SOLUTION                                  SOL00200
      INCLUDE MESAGE,LIST                                               SOL00300
      INCLUDE FILES,LIST                                                SOL00400
      INCLUDE MXBUFR,LIST                                               SOL00500
      INCLUDE NEWPRM,LIST                                               SOL00600
      INCLUDE WNDFLD,LIST                                               SOL00700
      INCLUDE WNDCOM,LIST                                               SOL00800
      INCLUDE WNDDTA,LIST                                               SOL00900
      INCLUDE WNDDTB,LIST                                               SOL01000
      INCLUDE TIMES,LIST                                                SOL01100
C     CALL INITIALIZATION ROUTINE                                       SOL01200
   10 CONTINUE                                                          SOL01300
      CALL MXSTRT                                                       SOL01400
      IF (TYPE3 .LT. 0) GO TO 30                                        SOL01500
C     IS SOLUTION SAME AS LAST CALCULATED                               SOL01600
      IF (KFLG1 .EQ. 9) GO TO 30                                        SOL01700
C     CALL SOLUTION                                                     SOL01800
      CALL MXSOLV                                                       SOL01900
      IF (IFLG .NE. -1) GO TO 20                                        SOL02000
      ICNT = ICNT+1                                                     SOL02100
      IF (ICNT .LT. 3) GO TO 10                                         SOL02200
      WRITE (XOTFIL,9000)                                               SOL02300
 9000 FORMAT ('0 *** CANNOT RECOVER FROM MODEL INSTABILITY. NO SOLUTION SOL02400
     1OUTPUT. PROG RETURNS CONTROL TO EXEC AND SETS TYPE3 = -1'/)       SOL02500
      TYPE3 = -1                                                        SOL02600
      GO TO 30                                                          SOL02700
   20 CONTINUE                                                          SOL02800
C     CALL OUTPUT                                                       SOL02900
      CALL MXLOUT                                                       SOL03000
      WRITE (XOTFIL,9001)                                               SOL03100
 9001 FORMAT ('0 *** MESOSCALE WIND FIELD SOLUTION COMPLETED AND OUTPUT SOL03200
     1TO SAVE FILE'/)                                                   SOL03300
   30 CONTINUE                                                          SOL03400
      RETURN                                                            SOL03500
      END                                                               SOL03600
```

```
      SUBROUTINE MXSTRT                                                 MXS00100
C     SUBROUTINE TO INITIALIZE THE 2-LAYER MESOSCALE MODEL              MXS00200
      INCLUDE MESAGE,LIST                                               MXS00300
      INCLUDE FILES,LIST                                                MXS00400
      INCLUDE MXBUFR,LIST                                               MXS00500
      INCLUDE NEWPRM,LIST                                               MXS00600
      INCLUDE WNDFLD,LIST                                               MXS00700
      INCLUDE WNDCOM,LIST                                               MXS00800
      INCLUDE TIMES,LIST                                                MXS00900
      EQUIVALENCE (G1,G0),(DTLMDA,DTLMDO),(NCNT,NCNTO),(ABLK,ABLKO),    MXS01000
     1(ISMOTH,ISMTHO)                                                   MXS01100
      IDM = IDP-1                                                       MXS01200
      JDM = JDP-1                                                       MXS01300
      IFLG = 0                                                          MXS01400
      IF (PN .GE. HMIN+ABLKN) GO TO 100                                 MXS01500
      WRITE (XOTFIL,9000) PN                                            MXS01600
      TYPE3 = -1                                                        MXS01700
      RETURN                                                            MXS01800
  100 CONTINUE                                                          MXS01900
C     INPUT THE TERRAIN ELEVATION DATA WHERE THERE ARE IDP*JDP POINTS   MXS02000
C     AND THE FIRST 5 (LAST 5) ROWS AND COLUMNS ARE EQUAL TO HMIN.      MXS02100
      CALL BLKIN(IDP*JDP,H,1,WSTRRN,ISTS)                               MXS02200
C     INPUT LAST SOLUTION                                               MXS02300
      NWRDS = 3*IDP*JDP+68                                              MXS02400
      CALL BLKIN(NWRDS,NETRAD,2,MXLYRF,ISTS)                            MXS02500
C     MOVE LAST SOLUTION DOWN FILE                                      MXS02600
      CALL BLKOUT(NWRDS,NETRAD,1,MXLYRF,ISTS)                           MXS02700
      ABLKO = ABLKN                                                     MXS02800
      DTLMDO = DTLMDN                                                   MXS02900
      GO = GN                                                           MXS03000
      ISMTHO = ISMTHN                                                   MXS03100
      NCNTO = NCNTN                                                     MXS03200
      NETRAD = NETRUN                                                   MXS03300
      XSOURC = XSOURN                                                   MXS03400
      YSOURC = YSOURN                                                   MXS03500
      ZSOURC = ZSOURN                                                   MXS03600
      DO 250 I=1,25                                                     MXS03700
      POTMPR(I) = POTMPN(I)                                             MXS03800
```

```
  250 ZELEVR(I) = ZELEVN(I)                                                 MXS03900
      SURFTA = SURFTN                                                       MXS04000
      A1 = PN-PO                                                            MXS04100
      A2 = UN-UO                                                            MXS04200
      A3 = VN-VO                                                            MXS04300
      A4 = PO-HMIN                                                          MXS04400
C     DETERMINE IF COLD OR HOT START                                        MXS04500
      IULIAN = JULIAN+DAY                                                   MXS04600
      IMONTH = MONTH                                                        MXS04700
      IDAY = DAY                                                            MXS04800
      IHOUR = HOUR                                                          MXS04900
      NINUTE = MINUTE                                                       MXS05000
      IF (ABS(A1)+ABS(A2)+ABS(A3)-.001) 400,400,270                        MXS05100
  270 CONTINUE                                                              MXS05200
      IF (KFLG1 .GT. 0) GO TO 300                                           MXS05300
      IFLG1 = 2                                                             MXS05400
C     ARE THE NEW INITIALIZATION PARAMETERS VELOCITY AND LAYER HEIGHT       MXS05500
C     WITHIN 25% OF OLD VALUE AND IS DIFF IN OLD AND NEW DIR <= 10 DEG     MXS05600
C     IF YES UPDATE OLD SOLUTION , ELSE COLD START                          MXS05700
      VL1 = UO                                                              MXS05800
      VR1 = VO                                                              MXS05900
      VL2 = UN                                                              MXS06000
      VR2 = VN                                                              MXS06100
      CALL HEC003(VL1,VR1)                                                  MXS06200
      CALL HEC003(VL2,VR2)                                                  MXS06300
      IF (0.75*VL1 .GT. VL2.OR.1.75*VL1 .LT. VL2) GO TO 300                MXS06400
      IF (.075*PO .GT. PN.OR.1.75*PO .LT. PN) GO TO 300                     MXS06500
      VL1 = ABS(VR1-VR2)                                                    MXS06600
      IF (VL1 .GT. 180.0) VL1 = 360.0-VL1                                   MXS06700
      IF (VL1 .LE. 10.0) GO TO 310                                          MXS06800
  300 CONTINUE                                                              MXS06900
      A1 = PN                                                               MXS07000
      A2 = UN                                                               MXS07100
      A3 = VN                                                               MXS07200
      A4 = -HMIN                                                            MXS07300
      IFLG1 = 1                                                             MXS07400
  310 CONTINUE                                                              MXS07500
      UO = UN                                                               MXS07600
```

```
      VO = VN
      PO = PN
      DO 360 J=1,JDP
      DO 360 I=1,IDP
      GO TO (320,330),IFLG1
  320 M(I,J) = 0.0
      N(I,J) = 0.0
      P(I,J) = 0.0
  330 A5 = P(I,J)-H(I,J)
      P(I,J) = A5+A1
      IF (P(I,J) .LT. ABLK) GO TO 350
  340 M(I,J) = A5*M(I,J)+A4*A2+UN*A1
      N(I,J) = A5*N(I,J)+A4*A3+VN*A1
      GO TO 360
C     MINIMUM DEPTH CONSTRAINT
  350 P(I,J) = ABLK
      M(I,J) = 0.0
      N(I,J) = 0.0
  360 CONTINUE
C     ARRAYS INITIALIZED
C     CALC TIME ITERATION LOOP CONSTANTS
      STPN = 1.0/((IDP-10)*(JDP-10))
      G2 = 0.5*G1
      DTL = DTLMDA*.70710678*XMIN
      NN = 0
      ISM = 0
      TIM = 0.0
      ICNT = NCNT
      DPMAX = 20000.
      UVMIN = 1.0E-19*SQRT(DPMAX/AMIN1(DXPI(1),DYPJ(1)))
      UVEQU = 1.0E-37/AMIN1(2.0*DELXI(1),2.0*DELYJ(1))
      UVMAX = AMAX1(10.0*ABS(UN),10.0*ABS(VN))
      IF (UVMAX .LT. 5.0) UVMAX = 5.0
      IF (IFLG1 .EQ. 1) WRITE (XOTFIL,9002)
      IF (IFLG1 .EQ. 2) WRITE (XOTFIL,9003)
      GO TO 410
  400 WRITE (XOTFIL,9001)
      KFLG1 = 9
```

```
      CALL BLKOUT(NWRDS,NETRAD,2,MXLYRF,ISTS)                           MXS11500
  410 CONTINUE                                                          MXS11600
 9000 FORMAT ('0 *-*-* ERROR - LAYER HEIGHT IS LESS THAN MINIMUM + ABLKNMXS11700
     1CONSTRAINT, PROG CANNOT CONTINUE, PN=',F7.2)                      MXS11800
 9001 FORMAT ('0 *-* MESOSCALE WINDFIELD SOLUTION SAME AS LAST SOLUTION,MXS11900
     1 PROG DOES NOT RECALC, PROG USES LAST SOLUTION *-*')              MXS12000
 9002 FORMAT ('0 *-* COLD START USED FOR MESOSCALE INITIALIZATION')     MXS12100
 9003 FORMAT ('0 *-*  HOT START USED FOR MESOSCALE INITIALIZATION')     MXS12200
      RETURN                                                            MXS12300
      END                                                               MXS12400
```

```
      SUBROUTINE MXSOLV                                                 MXV00100
C     ROUTINE TO SOLVE 2-LAYER MESOSCALE MODEL                          MXV00200
      INCLUDE FILES,LIST                                                MXV00300
      INCLUDE MXBUFR,LIST                                               MXV00400
      INCLUDE WNDFLD,LIST                                               MXV00500
      INCLUDE WNDCOM,LIST                                               MXV00600
      INCLUDE NEWPRM,LIST                                               MXV00700
      INCLUDE MESAGE,LIST                                               MXV00800
      EQUIVALENCE (G1,G0),(DTLMDA,DTLMD0),(NCNT,NCNT0),(ABLK,ABLK0),    MXV00900
     1(ISMOTH,ISMTH0)                                                   MXV01000
      FILTER(A,B,C,D,E,F,G,H,O) = A+0.125*(B+C+D+E-4.0*A)+0.0625*(F+G+H+MXV01100
     1O-4.0*A)                                                          MXV01200
C     BEGIN TIME ITERATION LOOP                                         MXV01300
      CALL MYPRG(1,1,J,IFLG,KFLG2)                                      MXV01400
C     SET TIME TO START TO CHECK FOR CONVERGENCE                        MXV01500
      TIMMIN = 1200.0                                                   MXV01600
      IF (IFLG1 .NE. 2) GO TO 170                                       MXV01700
      TIMMIN = 600                                                      MXV01800
      TIMMAX = 1800                                                     MXV01900
  170 CONTINUE                                                          MXV02000
      JDPM5 = JDP-5                                                     MXV02100
      IDPM5 = IDP-5                                                     MXV02200
  200 NN = NN+1                                                         MXV02300
C     FILTER COUNTER                                                    MXV02400
      IF (ISM .GE. ISMOTH) ISM = 0                                      MXV02500
      ISM = ISM+1                                                       MXV02600
C     DT RECALCULATION COUNTER                                          MXV02700
      IF (ICNT .GE. NCNT) ICNT = 0                                      MXV02800
      ICNT = ICNT+1                                                     MXV02900
      IF (ICNT .GT. 1) GO TO 300                                        MXV03000
C     CALC TIME STEP INCREMENT DT                                       MXV03100
      XMAX = -1.0E5                                                     MXV03200
      DO 220 J=1,JDP                                                    MXV03300
      DO 220 I=1,IDP                                                    MXV03400
      IF (P(I,J) .LE. 0.0) GO TO 220                                    MXV03500
      CHK = SQRT(M(I,J)*M(I,J)+N(I,J)*N(I,J))/P(I,J)+SQRT(ABS(G1*P(I,J))MXV03600
     1)                                                                 MXV03700
      XMAX = AMAX1(CHK,XMAX)                                            MXV03800
```

```
  220 CONTINUE                                            MXV03900
      DT = ABS(DTL/XMAX)                                  MXV04000
      IF (DT .LT. 1.0) DT = 1.0                           MXV04100
      A0 = 0.5*DT                                         MXV04200
  300 TIM = TIM+DT                                        MXV04300
      CALL MYPRG(2,I,J,IFLG,KFLG2)                        MXV04400
      D1 = 0.0                                            MXV04500
      D2 = 0.0                                            MXV04600
      D3 = 0.0                                            MXV04700
      D4 = 0.0                                            MXV04800
      JDPM5 = JDP-5                                       MXV04900
      IDPM5 = IDP-5                                       MXV05000
      DO 600 J=1,JDM                                      MXV05100
      DELYJJ = DELYJ(6)                                   MXV05200
      DLYJJ = DYPJ(6)                                     MXV05300
      IF (J .GT. 6) GO TO 350                             MXV05400
      IF (J .EQ. 1) GO TO 340                             MXV05500
      DELYJJ = DELYJ(J-1)                                 MXV05600
  340 DLYJJ = DYPJ(J)                                     MXV05700
      GO TO 360                                           MXV05800
  350 IF (J .LT. JDPM5) GO TO 360                         MXV05900
      DELYJJ = DELYJ(JDP-J)                               MXV06000
      DLYJJ = DYPJ(JDP-J)                                 MXV06100
  360 A1 = 2.0*DELYJJ                                     MXV06200
      A2 = 0.5*DELYJJ                                     MXV06300
      DO 600 I=1,IDM                                      MXV06400
      DELXII = DELXI(6)                                   MXV06500
      DLXII = DXPI(6)                                     MXV06600
      IF (I .GT. 6) GO TO 380                             MXV06700
      IF (I .EQ. 1) GO TO 370                             MXV06800
      DELXII = DELXI(I-1)                                 MXV06900
  370 DLXII = DXPI(I)                                     MXV07000
      GO TO 390                                           MXV07100
  380 IF (I .LT. IDPM5) GO TO 390                         MXV07200
      DELXII = DELXI(IDP-I)                               MXV07300
      DLXII = DXPI(IDP-I)                                 MXV07400
  390 CONTINUE                                            MXV07500
      PIJ = 1.0/P(I,J)                                    MXV07600
```

```
C     A3 = M(I+1,J+1)*N(I+1,J+1)/P(I+1,J+1)                       MXV08000
      A3 = (PIPJP*M(I+1,J+1))*N(I+1,J+1)                          MXV08100
      A4 = P(I,J)*P(I,J)                                          MXV08200
C     A5 = N(I,J)/P(I,J)                                          MXV08300
      A5 = N(I,J)*PIJ                                             MXV08400
      A6 = P(I+1,J+1)*P(I+1,J+1)                                  MXV08500
C     A7 = M(I,J+1)*N(I,J+1)/P(I,J+1)                             MXV08600
      A7 = (PIJP*M(I,J+1))*N(I,J+1)                               MXV08700
C     A8 = M(I,J)/P(I,J)                                          MXV08800
      A8 = M(I,J)*PIJ                                             MXV08900
C     A9 = M(I,J)*N(I,J)/P(I,J)                                   MXV09000
      A9 = A8*N(I,J)                                              MXV09100
C     A10 = M(I+1,J)**2/P(I+1,J)                                  MXV09200
      A10 = (PIPJ*M(I+1,J))*M(I+1,J)                              MXV09300
      A11 = P(I+1,J)*P(I+1,J)                                     MXV09400
C     A12 = M(I+1,J)*N(I+1,J)/P(I+1,J)                            MXV09500
      A12 = (PIPJ*M(I+1,J))*N(I+1,J)                              MXV09600
      A13 = P(I,J+1)*P(I,J+1)                                     MXV09700
C     A14 = N(I,J+1)**2/P(I,J+1)                                  MXV09800
      A14 = (PIJP*N(I,J+1))*N(I,J+1)                              MXV09900
C     A15 = M(I,J)*N(I,J)/P(I,J)**2                               MXV10000
      A15 = A9*PIJ                                                MXV10100
      IF (J .EQ. 1) GO TO 400                                     MXV10200
      A16 = G2*(P(I+1,J)+P(I,J))                                  MXV10300
      PIJM = 1.0/P(I,J-1)                                         MXV10400
      PIPJM = 1.0/P(I+1,J-1)                                      MXV10500
C     A17 = M(I,J+1)*N(I,J+1)/P(I,J+1)-M(I,J-1)*N(I,J-1)/P(I,J-1) MXV10600
      A17 = A7-(PIJM*M(I,J-1))*N(I,J-1)                           MXV10700
C     A18 = M(I+1,J)/P(I+1,J)                                     MXV10800
      A18 = M(I+1,J)*PIPJ                                         MXV10900
C     A19 = N(I,J+1)**2/P(I,J+1)-N(I,J-1)**2/P(I,J-1)             MXV11000
      A19 = A14-(PIJM*N(I,J-1))*N(I,J-1)                          MXV11100
C     A20 = P(I,J+1)**2-P(I,J-1)**2                               MXV11200
      A20 = A13-P(I,J-1)*P(I,J-1)                                 MXV11300
      A21 = H(I,J+1)-H(I,J-1)                                     MXV11400
```

```
      A22 = N(I,J+1)-N(I,J-1)                                           MXV11500
      A23 = ((M(I+1,J)-M(I,J))*DLXII+A2*(N(I+1,J+1)-N(I+1,J-1)+A22))   MXV11600
C     A24 = M(I+1,J)/P(I+1,J)+M(I,J)/P(I,J)                            MXV11700
      A24 = A18 + A8                                                    MXV11800
      FM = FP                                                           MXV11900
      FP = A2*(A3-(PIPJM*M(I+1,J-1))*N(I+1,J-1)+A17)+DLXII*((A10-M(I,J  MXV12000
     1)*A8+G2*(A11-A4))+A16*(H(I+1,J)-H(I,J)))                          MXV12100
      C2M = C2P                                                         MXV12200
      C2P = 0.5*((N(I+1,J)*PIPJ+A5)*FP+A24*(DLXII*(A12-A9)+A2*((PIPJP*  MXV12300
     1N(I+1,J+1))*N(I+1,J+1)-(PIPJM*N(I+1,J-1))*N(I+1,J-1)+A19+G2*(A6-P(MXV12400
     2I+1,J-1)*P(I+1,J-1)+A20))+A16*A2*(H(I+1,J+1)-H(I+1,J-1)+A21))-(A12MXV12500
     3*PIPJ+A15)*A23)                                                   MXV12600
      B2M = B2P                                                         MXV12700
      B2P = FP*A24+A23*(A16-0.5*(A18*A18+A8*A8))                        MXV12800
  400 IF (I .EQ. 1) GO TO 600                                           MXV12900
      A25 = 0.5*DELXII                                                  MXV13000
      PIMJ = 1.0/P(I-1,J)                                               MXV13100
      PIMJP = 1.0/P(I-1,J+1)                                            MXV13200
C     A26 = M(I+1,J)*N(I+1,J)/P(I+1,J)-M(I-1,J)*N(I-1,J)/P(I-1,J)      MXV13300
      A26 = A12-(PIMJ*M(I-1,J))*N(I-1,J)                                MXV13400
      A27 = G2*(P(I,J+1)+P(I,J))                                        MXV13500
      A28 = N(I,J+1)*PIJP                                               MXV13600
C     A29 = M(I+1,J)**2/P(I+1,J)-M(I-1,J)**2/P(I-1,J)                  MXV13700
      A29 = A10-(PIMJ*M(I-1,J))*M(I-1,J)                                MXV13800
C     A30 = P(I+1,J)**2-P(I-1,J)**2                                    MXV13900
      A30 = A11-P(I-1,J)*P(I-1,J)                                       MXV14000
      A31 = H(I+1,J)-H(I-1,J)                                           MXV14100
      A32 = M(I+1,J)-M(I-1,J)                                           MXV14200
      A33 = ((N(I,J+1)-N(I,J))*DLYJJ+A25*(M(I+1,J+1)-M(I-1,J+1)+A32))   MXV14300
C     A34 = N(I,J+1)/P(I,J+1)+N(I,J)/P(I,J)                            MXV14400
      A34 = A28+A5                                                      MXV14500
      GM = GP(I-1)                                                      MXV14600
      GP(I-1) = A25*(A3-(PIMJP*M(I-1,J+1))*N(I-1,J+1)+A26)+DLYJJ*((A14  MXV14700
     1-N(I,J)*A5+G2*(A13-A4))+A27*(H(I,J+1)-H(I,J)))                    MXV14800
      B3M = B3P(I-1)                                                    MXV14900
      B3P(I-1) = 0.5*((M(I,J+1)*PIJP+A8)*GP(I-1)+A34*(DLYJJ*(A7-A9)  +A2MXV15000
     15*((PIPJP*M(I+1,J+1))*M(I+1,J+1)-(PIMJP*M(I-1,J+1))*M(I-1,J+1)+A29MXV15100
     2+G2*(A6-P(I-1,J+1)*P(I-1,J+1)+A30))+A27*A25*(H(I+1,J+1)-H(I-1,J+1)MXV15200
```

```
     3+A31))-(A7*PIJP+A15)*A33)                                     MXV15300
      C3M = C3P(I-1)                                                MXV15400
      C3P(I-1) = GP(I-1)*A34+A33*(A27-0.5*(A28*A28+A5*A5))          MXV15500
  500 IF (I .EQ. 1.OR.J .EQ. 1) GO TO 600                           MXV15600
      A35 = 2.0*DELXII                                              MXV15700
      A36 = G2*P(I,J)                                               MXV15800
      A37 = DELXII*A32+DELYJJ*A22                                   MXV15900
      B1 = (DELXII*(A29+G2*A30)+DELYJJ*A17+A36*A35*A31)             MXV16000
      C1 = (DELYJJ*(A19+G2*A20)+DELXII*A26+A36*A1*A21)              MXV16100
C     CHANGE IN M                                                   MXV16200
      A34 = B2P-B2M                                                 MXV16300
      IF (ABS(A34) .LT. UVEGU) A34 = 0.0                            MXV16400
      A33 = B3P(I-1)-B3M                                            MXV16500
      IF (ABS(A33) .LT. UVEGU) A33 = 0.0                            MXV16600
      E1 = DT*(-B1+A0*(A35*A34+A1*A33+G1*DELXII*A31*A37))           MXV16700
C     CHANGE IN N                                                   MXV16800
      A34 = C2P-C2M                                                 MXV16900
      IF (ABS(A34) .LT. UVEGU) A34 = 0.0                            MXV17000
      A33 = C3P(I-1)-C3M                                            MXV17100
      IF (ABS(A33) .LT. UVEGU) A33 = 0.0                            MXV17200
      E2 = DT*(-C1+A0*(A35*A34+A1*A33+G1*DELYJJ*A21*A37))           MXV17300
C     CHANGE IN P                                                   MXV17400
      A34 = FP-FM                                                   MXV17500
      IF (ABS(A34) .LT. UVEGU) A34 = 0.0                            MXV17600
      A33 = GP(I-1)-GM                                              MXV17700
      IF (ABS(A33) .LT. UVEGU) A33 = 0.0                            MXV17800
      E3 = DT*(-A37+A0*(A35*A34+A1*A33))                            MXV17900
      M(I-1,J-1) = M(I,J)+E1                                        MXV18000
      N(I-1,J-1) = N(I,J)+E2                                        MXV18100
      P(I-1,J-1) = P(I,J)+E3                                        MXV18200
      IF (ABS(M(I-1,J-1)) .LT. UVMIN) M(I-1,J-1) = 0.0              MXV18300
      IF (ABS(N(I-1,J-1)) .LT. UVMIN) N(I-1,J-1) = 0.0              MXV18400
      CALL MYPRG(3,I,J,IFLG,KFLG2)                                  MXV18500
      IF (ISMOTH.GT.0.AND.ISM.NE.ISMOTH.AND.KK.NE.0) GO TO 600      MXV18600
      IF (I .LT. 6.OR.I .GT. IDPM5) GO TO 600                       MXV18700
      IF (J .LT. 6.OR.J .GT. JDPM5) GO TO 600                       MXV18800
      D1 = D1+P(I-1,J-1)                                            MXV18900
      D2 = D2+P(I-1,J-1)*P(I-1,J-1)                                 MXV19000
```

```
      D3 = D3+E3                                                        MXV19100
      D4 = D4+E3*E3                                                     MXV19200
  600 CONTINUE                                                          MXV19300
      CALL MYPRG(4,I,J,IFLG,KFLG2)                                      MXV19400
      DO 720 J=2,JDM                                                    MXV19500
      L = JDP-J                                                         MXV19600
      IF (ISM .NE. ISMOTH) GO TO 700                                    MXV19700
      L1 = L-1                                                          MXV19800
      L2 = L+1                                                          MXV19900
      IF (L1 .LT. 1) L1 = 1                                             MXV20000
      IF (L2 .GT. JDM-1) L2 = JDM-1                                     MXV20100
  700 DO 720 I=2,IDM                                                    MXV20200
      K = IDP-I                                                         MXV20300
      IF (ISM .EQ. ISMOTH) GO TO 710                                    MXV20400
      M(K+1,L+1) = M(K,L)                                               MXV20500
      N(K+1,L+1) = N(K,L)                                               MXV20600
      P(K+1,L+1) = P(K,L)                                               MXV20700
      GO TO 720                                                         MXV20800
  710 K1 = K-1                                                          MXV20900
      K2 = K+1                                                          MXV21000
      IF (K1 .LT. 1) K1 = 1                                             MXV21100
      IF (K2 .GT. IDM-1) K2 = IDM-1                                     MXV21200
      M(K+1,L+1) = FILTER(M(K,L),M(K2,L),M(K1,L),M(K,L2),M(K,L1),M(K1,L1MXV21300
     1),M(K1,L2),M(K2,L1),M(K2,L2))                                     MXV21400
      N(K+1,L+1) = FILTER(N(K,L),N(K2,L),N(K1,L),N(K,L2),N(K,L1),N(K1,L1MXV21500
     1),N(K1,L2),N(K2,L1),N(K2,L2))                                     MXV21600
      P(K+1,L+1) = FILTER(P(K,L),P(K2,L),P(K1,L),P(K,L2),P(K,L1),P(K1,L1MXV21700
     1),P(K1,L2),P(K2,L1),P(K2,L2))                                     MXV21800
  720 CONTINUE                                                          MXV21900
C     FIX BOUNDARY                                                      MXV22000
      DO 730 J=2,JDM                                                    MXV22100
      M(1,J) = M(2,J)                                                   MXV22200
      N(1,J) = N(2,J)                                                   MXV22300
      P(1,J) = P(2,J)                                                   MXV22400
      M(IDP,J) = M(IDM,J)                                               MXV22500
      N(IDP,J) = N(IDM,J)                                               MXV22600
  730 P(IDP,J) = P(IDM,J)                                               MXV22700
      DO 740 I=1,IDP                                                    MXV22800
```

```
      M(I,1) = M(I,2)
      N(I,1) = N(I,2)
      P(I,1) = P(I,2)
      M(I,JDP) = M(I,JDM)
      N(I,JDP) = N(I,JDM)
  740 P(I,JDP) = P(I,JDM)
C     CHECK FOR MINIMUM, MAXIMUM DEPTH AND VELOCITY VIOLATIONS
      DO 1400 J=1,JDP
      DO 1400 I=1,IDP
      IF (P(I,J) .GE. ABLK) GO TO 1350
      M(I,J) = 0.0
      N(I,J) = 0.0
      P(I,J) = ABLK
 1350 CONTINUE
      IF (ABS(M(I,J)) .LE. UVMAX*P(I,J)) GO TO 1360
      E3 = UVMAX*P(I,J)
      IF (M(I,J) .LT. 0.0) E3 = -E3
      M(I,J) = E3
 1360 IF (ABS(N(I,J)) .LE. UVMAX*P(I,J)) GO TO 1370
      E3 = UVMAX*P(I,J)
      IF (N(I,J) .LT. 0.0) E3 = -E3
      N(I,J) = E3
 1370 CONTINUE
      IF (P(I,J) .GT. DPMAX) P(I,J) = DPMAX
 1400 CONTINUE
C     CHECK FOR CONVERGENCE OF SOLUTION
      IF (ISMOTH .EQ. 0) GO TO 1600
      IF (ISM .NE. ISMOTH) GO TO 1620
 1600 CONTINUE
      D4 = (D4-D3*D3*STPN)/(D2-D1*D1*STPN)
      CHK = 0.0
      IF (TIM .LT. TIMMIN) GO TO 1610
      SIX = SIX+TIM
      SIY = SIY+D4
      SIXY = SIXY+D4*TIM
      SIXX = SIXX+TIM*TIM
      KK = KK+1
      IF (KK .EQ. 1) GO TO 1610
```

```
      CHK = (SIX*SIY-KK*SIXY)/(SIX*SIX-KK*SIXX)                         MXV26700
      IF (CHK .GT. 0.0) GO TO 1610                                      MXV26800
      IF (D4LAST-D4 .GT. 1.0E-5) GO TO 1610                             MXV26900
      IF (D4LAST-D4 .LT. 0.0) GO TO 1610                                MXV27000
C     SOLUTION CONVERGENCE REACHED                                      MXV27100
      IFLG = 1                                                          MXV27200
 1610 CONTINUE                                                          MXV27300
      D4LAST = D4                                                       MXV27400
 1620 CONTINUE                                                          MXV27500
      CALL MYPRG(5,I,J,IFLG,KFLG2)                                      MXV27600
      IF (IFLG .EQ. 1) GO TO 1630                                       MXV27700
      IF (TIM .LT. TIMMAX) GO TO 200                                    MXV27800
      IF (ISMOTH .EQ. 0) GO TO 1630                                     MXV27900
      IF (ISM .NE. ISMOTH) GO TO 200                                    MXV28000
 1630 CONTINUE                                                          MXV28100
      CALL MYPRG(6,I,J,IFLG,KFLG2)                                      MXV28200
      RETURN                                                            MXV28300
      END                                                               MXV28400
```

```
      SUBROUTINE MXLOUT                                            MXL00100
      INCLUDE FILES,LIST                                           MXL00200
      INCLUDE MXBUFR,LIST                                          MXL00300
      INCLUDE WNDCOM,LIST                                          MXL00400
      INCLUDE WNDFLD,LIST                                          MXL00500
      DO 100 J=1,JDP                                               MXL00600
      DO 100 I=1,IDP                                               MXL00700
      DTL = 1.0/P(I,J)                                             MXL00800
      M(I,J) = M(I,J)*DTL                                          MXL00900
      N(I,J) = N(I,J)*DTL                                          MXL01000
  100 P(I,J) = P(I,J)+H(I,J)                                       MXL01100
      NOUT = 3*IDP*JDP+68                                          MXL01200
      CALL BLKOUT(NOUT,NETRAD,2,MXLYRF,ISTS)                       MXL01300
      RETURN                                                       MXL01400
      END                                                          MXL01500
```

```
      SUBROUTINE MYPRG(ILFQ,I,J,JFLG,KFLG2)                             MYP00100
      INCLUDE MXBUFR,LIST                                               MYP00200
      INCLUDE WNDFLD,LIST                                               MYP00300
      INCLUDE WNDCOM,LIST                                               MYP00400
      INCLUDE FILES,LIST                                                MYP00500
      EQUIVALENCE (ISMTHO,ISMOTH)                                       MYP00600
      DATA RAD/.01745/                                                  MYP00700
      EQUIVALENCE (ABLK,ABLKO),(DTLMDA,DTLMDO),(NCNT,NCNTO),(G1,GO)     MYP00800
      COMMON /SWELL/ PLTM(200),VAR(200,2),VOL(200,2)                    MYP00900
      DIMENSION PRINT(4),OUTR(1),OUTS(1),OUTT(1)                        MYP01000
      EQUIVALENCE (OUTR,M),(OUTS,PLTM),(OUTT,H)                         MYP01100
      DIMENSION X(41),Y(41)                                             MYP01200
      DATA PRINT/30.,60.,90.,120./,ITSTP/200/                           MYP01300
C     BEGIN TIME ITERATION LOOP                                         MYP01400
      GO TO (8031,8032,8033,8034,8035,8036),ILFQ                        MYP01500
 8031 CONTINUE                                                          MYP01600
      IST = 6                                                           MYP01700
      JST = 6                                                           MYP01800
      IND = IDP-5                                                       MYP01900
      JND = JDP-5                                                       MYP02000
      NX = IND-JST+1                                                    MYP02100
      NY = JND-JST+1                                                    MYP02200
      E0 = 1.0/(NX*NY)                                                  MYP02300
      II = IST+1                                                        MYP02400
      JJ = JST+1                                                        MYP02500
      X(6) = XSTRT                                                      MYP02600
      X(5) = X(6)-2.0*XINC                                              MYP02700
      X(4) = X(5)-4.0*XINC                                              MYP02800
      X(3) = X(4)-8.0*XINC                                              MYP02900
      X(2) = X(3)-16.0*XINC                                             MYP03000
      X(1) = X(2)-32.0*XINC                                             MYP03100
      DO 6 I=II,IND                                                     MYP03200
    6 X(I) = X(I-1)+XINC                                                MYP03300
      X(IND+1) = X(IND)+2.0*XINC                                        MYP03400
      X(IND+2) = X(IND+1)+4.0*XINC                                      MYP03500
      X(IND+3) = X(IND+2)+8.0*XINC                                      MYP03600
      X(IND+4) = X(IND+3)+16.0*XINC                                     MYP03700
      X(IND+5) = X(IND+4)+32.0*XINC                                     MYP03800
```

```
      Y(6) = YSTRT                                                        MYP03900
      Y(5) = Y(6)-2.0*YINC                                                MYP04000
      Y(4) = Y(5)-4.0*YINC                                                MYP04100
      Y(3) = Y(4)-8.0*YINC                                                MYP04200
      Y(2) = Y(3)-16.0*YINC                                               MYP04300
      Y(1) = Y(2)-32.0*YINC                                               MYP04400
      DO 7 J=JJ,JND                                                       MYP04500
    7 Y(J) = Y(J-1)+YINC                                                  MYP04600
      Y(JND+1) = Y(JND)+2.0*YINC                                          MYP04700
      Y(JND+2) = Y(JND+1)+4.0*YINC                                        MYP04800
      Y(JND+3) = Y(JND+2)+8.0*YINC                                        MYP04900
      Y(JND+4) = Y(JND+3)+16.0*YINC                                       MYP05000
      Y(JND+5) = Y(JND+4)+32.0*YINC                                       MYP05100
      MAXLL = 4                                                           MYP05200
      LL = 1                                                              MYP05300
      IN = 1                                                              MYP05400
      LINES = 60                                                          MYP05500
      RETURN                                                              MYP05600
C*C*C                                                                     MYP05700
 8032 CONTINUE                                                            MYP05800
      IF (ISW1 .NE. 0) GO TO 330                                          MYP05900
  310 ISW1 = 1                                                            MYP06000
C     CALC INITIAL MOMENTUM AND VOLUME                                    MYP06100
      PLTM(1) = 0.0                                                       MYP06200
      VAR(1,1) = 0.0                                                      MYP06300
      VAR(1,2) = 0.0                                                      MYP06400
      E1 = 0.0                                                            MYP06500
      E3 = 0.0                                                            MYP06600
      DO 320 J=2,JDM                                                      MYP06700
      E4 = 0.25*(Y(J+1)-Y(J-1))                                           MYP06800
      DO 320 I=2,IDM                                                      MYP06900
      B1 = E4*(X(I+1)-X(I-1))                                             MYP07000
      E1 = E1+B1*SQRT(M(I+1,J+1)*M(I+1,J+1)+N(I+1,J+1)*N(I+1,J+1))        MYP07100
      E3 = E3+P(I+1,J+1)*B1                                               MYP07200
  320 CONTINUE                                                            MYP07300
      VOL(1,1) = E1                                                       MYP07400
      VOL(1,2) = E3                                                       MYP07500
  330 CONTINUE                                                            MYP07600
```

```
      E5 = 0.0                                                          MYP07700
      E6 = 0.0                                                          MYP07800
      E7 = 0.0                                                          MYP07900
      E8 = 0.0                                                          MYP08000
      E9 = 0.0                                                          MYP08100
      E10 = 0.0                                                         MYP08200
      E11 = 0.0                                                         MYP08300
      E12 = 0.0                                                         MYP08400
      RETURN                                                            MYP08500
C*C*C                                                                   MYP08600
 8033 CONTINUE                                                          MYP08700
      IF (I .LT. IST.OR.I .GT. IND) GO TO 600                           MYP08800
      IF (J .LT. JST.OR.J .GT. JND) GO TO 600                           MYP08900
C     CALCULATE REL VAR OF MOMENTUM AND DEPTH                           MYP09000
      E1 = 0.25*(X(I+1)-X(I-1))*(Y(J+1)-Y(J-1))                         MYP09100
      E3 = SQRT(M(I-1,J-1)*M(I-1,J-1)+N(I-1,J-1)*N(I-1,J-1))            MYP09200
      E4 = E1*E3                                                        MYP09300
      E9 = E9+E4                                                        MYP09400
      E11 = E11+E4*E4                                                   MYP09500
      E3 = E3/P(I-1,J-1)                                                MYP09600
      E10 = E10+P(I-1,J-1)                                              MYP09700
      E12 = E12+P(I-1,J-1)*P(I-1,J-1)                                   MYP09800
      E2 = SQRT(M(I,J)*M(I,J)+N(I,J)*N(I,J))                            MYP09900
      E4 = E4-E1*E2                                                     MYP10000
      E5 = E5+E4                                                        MYP10100
      E7 = E7+E4*E4                                                     MYP10200
      E3 = P(I-1,J-1)-P(I,J)                                            MYP10300
      E6 = E6+E3                                                        MYP10400
      E8 = E8+E3*E3                                                     MYP10500
  600 CONTINUE                                                          MYP10600
      RETURN                                                            MYP10700
C*C*C                                                                   MYP10800
 8034 CONTINUE                                                          MYP10900
      E1 = (E7-E5*E5*E0)                                                MYP11000
      E2 = (E8-E6*E6*E0)                                                MYP11100
      E3 = E1/(E11-E9*E9*E0)                                            MYP11200
      E4 = E2/(E12-E10*E10*E0)                                          MYP11300
C     CALC MOMENTUM AND VOLUME                                          MYP11400
```

```
      E5 = 0.0                                                          MYP11500
      E7 = 0.0                                                          MYP11600
      DO 610 J=2,JDM                                                    MYP11700
      E8 = 0.25*(Y(J+1)-Y(J-1))                                         MYP11800
      DO 610 I=2,IDM                                                    MYP11900
      E9 = E8*(X(I+1)-X(I-1))                                           MYP12000
      E5 = E5+E9*SQRT(M(I,J)*M(I,J)+N(I,J)*N(I,J))                      MYP12100
      E7 = E7+P(I,J)*E9                                                 MYP12200
  610 CONTINUE                                                          MYP12300
      E1 = E5                                                           MYP12400
      E2 = E7                                                           MYP12500
      IN = IN+1                                                         MYP12600
      IF (IN .GT. 200) GO TO 620                                        MYP12700
      PLTM(IN) = TIM                                                    MYP12800
      VAR(IN,1) = E3                                                    MYP12900
      VAR(IN,2) = E4                                                    MYP13000
      VOL(IN,1) = E1                                                    MYP13100
      VOL(IN,2) = E2                                                    MYP13200
      IF (NN .GT. 1) GO TO 620                                          MYP13300
      E5 = VOL(1,1)                                                     MYP13400
      E6 = VOL(1,2)                                                     MYP13500
      E7 = 0.0                                                          MYP13600
      IF (KFLG2 .EQ. 0) GO TO 630                                       MYP13700
      WRITE (XOTFIL,2001)                                               MYP13800
      WRITE (XOTFIL,2000) JFLG,E5,E6,E7,E7,E7                           MYP13900
      LINES = 1                                                         MYP14000
  620 IF (KFLG2 .EQ. 0) GO TO 630                                       MYP14100
      IF (LINES .GT. 57) WRITE (XOTFIL,2001)                            MYP14200
      LINES = 1                                                         MYP14300
      WRITE (XOTFIL,2000) ISM,E1,E2,E3,E4,TIM                           MYP14400
  630 CONTINUE                                                          MYP14500
      RETURN                                                            MYP14600
C*C*C                                                                   MYP14700
 8035 CONTINUE                                                          MYP14800
      LLST = LL                                                         MYP14900
C     OUTPUT CALCULATIONS IF CORRECT TIME STEP                          MYP15000
      IF (JFLG .EQ. 1) GO TO 1410                                       MYP15100
      IF (PRINT(LL)*60.0 .GT. TIM) GO TO 1700                           MYP15200
```

```
      IF (ISMOTH .EQ. 0) GO TO 1405
      IF (ISM .NE. ISMOTH) GO TO 1700
 1405 LL = LL+1
 1410 IS1 = NETRAD
      NETRAD = JFLG
      TS1 = SURFTA
      IS2 = NCNTO
      SURFTA = TIM
      NCNTO = NN
      IOF = IOF +1
      CALL BLKOUT(5111,NETRAD,IOF,SPARE,ISTS)
      NETRAD = IS1
      SURFTA = TS1
      NCNTO = IS2
      IF (KFLG2 .GT. 0) WRITE (XOTFIL,2007) IOF
 1700 CONTINUE
      RETURN
C*C*C
 8036 CONTINUE
C     END OF ITERATIONS
      IF (IN .GT. 200) IN = 200
      IS1 = NETRAD
      NETRAD = -IN
      IOF = IOF+1
      DO 1810 I=1,1000
 1810 OUTT(I) = OUTR(I)
      DO 1820 I=1,1000
 1820 OUTR(I) = OUTS(I)
      CALL BLKOUT(5111,NETRAD,IOF,SPARE,ISTS)
      NETRAD = IS1
      DO 1830 I=1,1000
 1830 OUTR(I) = OUTT(I)
      CALL BLKIN(1681,H,1,WSTRRN,ISTS)
      WRITE (XOTFIL,2002) IOF
      IF (KFLG2 .EQ. 0) GO TO 1850
      IF (JFLG .EQ. 1) GO TO 1840
      WRITE (XOTFIL,2003) NN,TIM
      GO TO 1850
```

```
 1840 WRITE (XOTFIL,2004) NN,TIM                                     MYP19100
 1850 CONTINUE                                                       MYP19200
      RETURN                                                         MYP19300
 2000 FORMAT (' ISM=',I3,' MOMENTUM=',E12.6,' /VOLUME=',E12.6,' /REL VARMYP19400
     1 MOMENTUM=',E12.6,' /REL VAR DEPTH=',E12.6,' /TIME=',F10.4)    MYP19500
 2001 FORMAT ('1')                                                   MYP19600
 2002 FORMAT ('0',20X,'*** ANALYSIS - MESO-SCALE SOLUTION COMPLETED, ',IMYP19700
     12,' BLOCKS WRITTEN TO FILE SPARE ***')                         MYP19800
 2003 FORMAT (' * SOLUTION STOPPED AT TIME STEP ',I3,' MODEL TIME ',F8.3MYP19900
     1,' SECONDS')                                                   MYP20000
 2004 FORMAT (' * SOLUTION CONVERGED AT TIME STEP ',I3,' MODEL TIME ',F8MYP20100
     1.3,' SECONDS')                                                 MYP20200
 2007 FORMAT ('0 BLOCK ',I6,' WRITTEN TO TEMP FILE SPARE')           MYP20300
      END                                                            MYP20400
```

```
      SUBROUTINE TRYDIF                                                 TRY00100
      INCLUDE MESAGE,LIST                                               TRY00200
C     CALCULATE TRAJECTORY AND ASSOCIATED PARAMETERS                    TRY00300
      INCLUDE DIFUZZ,LIST                                               TRY00400
      INCLUDE COMWND,LIST                                               TRY00500
      INCLUDE WNDDTB,LIST                                               TRY00600
      INCLUDE SOURCE,LIST                                               TRY00700
      INCLUDE FILES,LIST                                                TRY00800
      DATA GAMMA/-1.0/                                                  TRY00900
      DATA KINDSO,SIGXO,SIGYO,SIGZO/0,3*-1.0/                           TRY01000
      NAMELIST /QLST2/ XSOUR,YSOUR,Q,SIGXO,SIGYO,SIGZO,H,Z,TAU,ALPHA,   TRY01100
     1BETA,ZSOUR,KINDSO,DECAY,BLAMDA,GAMMA,STTEMP,RADIUS,EXTVEL,VS,     TRY01200
     2CONISS,DOSISS,TEMPSF,RTLAPS,KFLG1,KFLG2                           TRY01300
      IF (TYPE3 .GT. 0) READ (XINFIL,QLST2)                             TRY01400
      IF (TYPE3 .LT. 0) KINDSO = IABS(TYPE3)                            TRY01500
      TYPE3 = IABS(TYPE3)                                               TRY01600
      IF (KINDSO .NE. 0) GO TO 10                                       TRY01700
      IF (ALPHA .LE. 0.0) ALPHA = 1.0                                   TRY01800
      IF (BETA .LE. 0.0) BETA = 1.0                                     TRY01900
      IF (Q .LE. 0.0) Q = 1.0                                           TRY02000
      IF (TAU .LE. 0.0) TAU = 2.5                                       TRY02100
      GO TO 20                                                          TRY02200
   10 IF (ALPHA .LE. 0.0) ALPHA = 0.9                                   TRY02300
      IF (BETA .LE. 0.0) BETA = 1.0                                     TRY02400
      IF (KINDSO .EQ. 2) GO TO 15                                       TRY02500
      IF (Q .LE. 0.0) Q = 2520.0                                        TRY02600
      IF (STTEMP .LE. 0.0) STTEMP = 366.3                               TRY02700
      IF (RADIUS .LE. 0.0) RADIUS = 4.877                               TRY02800
      IF (EXTVEL .LE. 0.0) EXTVEL = 2.1476                              TRY02900
      IF (H .LE. 0.0) H = 252.4                                         TRY03000
      IF (XSOUR .LE. 0.0) XSOUR = 356000.0                              TRY03100
      IF (YSOUR .LE. 0.0) YSOUR = 3520100.0                             TRY03200
      IF (ZSOUR .LE. 0.0) ZSOUR = 1150.6                                TRY03300
      GO TO 20                                                          TRY03400
   15 IF (Q .LE. 0.0) Q = 1764.0                                        TRY03500
      IF (STTEMP .LE. 0.0) STTEMP = 352.4                               TRY03600
      IF (RADIUS .LE. 0.0) RADIUS = 3.658                               TRY03700
      IF (EXTVEL .LE. 0.0) EXTVEL = 2.7512                              TRY03800
```

```
      IF (H .LE. 0.0) H = 186.5                                         TRY03900
      IF (XSOUR .LE. 0.0) XSOUR = 356000.0                              TRY04000
      IF (YSOUR .LE. 0.0) YSOUR = 3520100.0                             TRY04100
      IF (ZSOUR .LE. 0.0) ZSOUR = 1150.6                                TRY04200
   20 CONTINUE                                                          TRY04300
      IF (GAMMA .LT. 0.0) GAMMA = 1.0                                   TRY04400
      IF (CONISS(1) .GT. 0.0) GO TO 30                                  TRY04500
      CONISS(1) = 1.0E-4                                                TRY04600
      CONISS(2) = 1.0E-6                                                TRY04700
      CONISS(3) = 1.0E-12                                               TRY04800
   30 IF (DOSISS(1) .GT. 0.0) GO TO 40                                  TRY04900
      DOSISS(1) = 1.0E-4                                                TRY05000
      DOSISS(2) = 1.0E-6                                                TRY05100
      DOSISS(3) = 1.0E-12                                               TRY05200
   40 CONTINUE                                                          TRY05300
      IF (KFLG2 .GT. 0) WRITE (XOTFIL,QLST2)                            TRY05400
      CALL TRNPRT                                                       TRY05500
      IF (TYPE3 .LT. 0) RETURN                                          TRY05600
C     CALCULATE DIFFUSION                                               TRY05700
      CALL DIFFUZ                                                       TRY05800
      RETURN                                                            TRY05900
      END                                                               TRY06000
```

```
      SUBROUTINE TRNPRT                                                 TRN00100
      INCLUDE DIFUZZ,LIST                                               TRN00200
      INCLUDE COMWND,LIST                                               TRN00300
      INCLUDE WNDDTB,LIST                                               TRN00400
      INCLUDE FILES,LIST                                                TRN00500
      INCLUDE SOURCE,LIST                                               TRN00600
      REAL MM,NN                                                        TRN00700
C     DEFINE AN ARRAY STOR FROM XX(1) IN COMMON DIFUZZ TO END OF COMMON TRN00800
C     TO PROVIDE AN EQUIVALENCE INDEX. STOR(9) IS MM OR THE 13 TH       TRN00900
C     VARIABLE IN COMMON. STOR(1690) IS NN OR 41X41 PNTS. AWAY,         TRN01000
C     STOR(3371) IS PP OR 41X41 PNTS. AWAY.                             TRN01100
      DIMENSION STOR(5000),MM(41,41),NN(41,41),PP(41,41)                TRN01200
      EQUIVALENCE (STOR(1),XX(1))                                       TRN01300
C     SET STORAGE ADDRESSES FOR SOLUTION ARRAYS OBTAINED FROM SAVE FILE TRN01400
C     MXLYRF                                                            TRN01500
      EQUIVALENCE (STOR(9),MM(1,1)),(STOR(1690),NN(1,1)),(STOR(3371),PP(TRN01600
     11,1))                                                             TRN01700
C     DEFINE ARRAYS U,V AND W TO HOLD INTERNAL 31X31 REGION OF ARRAYS   TRN01800
C     MM,MM AND PP, ALSO X AND Y AXES ARRAYS                            TRN01900
      DIMENSION U(31,31),V(31,31),W(31,31),X(31),Y(31)                  TRN02000
C     SET STORAGE ADDRESSES FOR U,V AND W ARRAYS (INTERNAL 31X31 PNTS OFTRN02100
C     MM,NN AND PP) AT UPPER END OF COMMON DIFUZZ                       TRN02200
      EQUIVALENCE (STOR(2169),U(1,1)),(STOR(3130),V(1,1)),(STOR(4091),W(TRN02300
     11,1))                                                             TRN02400
C     SET STORAGE ADDRESS FOR X AND Y AXES OF 31X31 INTERNAL GRID       TRN02500
      EQUIVALENCE (R(1),X),(R(32),Y)                                    TRN02600
C     DEFINE WHERE IB,NX AND NY COME FROM                               TRN02700
      EQUIVALENCE (IB,HMIN),(NX,IDP),(NY,JDP)                           TRN02800
C     NO. OF PNTS. IN TRAJECTORY IB IS PASSED TO SUB DIFFUZ THROUGH HMINTRN02900
C     REMAINING EQUIVALENCE IS TO SAVE STORAGE                          TRN03000
      EQUIVALENCE (K1,R(66)),                                           TRN03100
     1(UVMX,R(67)),(DX,R(68)),(UUL,R(69)),(VVL,R(70)),(HXYM,R(71)),     TRN03200
     2(HXM,R(72)),(HX,R(73)),(HY,R(74)),(HX1,R(75)),(HY1,R(76)),        TRN03300
     3(HX5,R(77)),(HY5,R(78)),(HX21,R(79)),(HY21,R(80)),(HXY4,R(81)),   TRN03400
     4(U1,R(82)),(V1,R(83)),(U2,R(84)),(V2,R(85)),(XD,R(86)),(DT,R(87)),TRN03500
     5(T,R(88)),(XO,R(89)),(YO,R(90)),(N,R(91)),(K,R(92)),(NT,R(93)),   TRN03600
     6(X1,R(94)),(Y1,R(95)),(UX,R(96)),(UY,R(97)),(UXX,R(98)),          TRN03700
     7(UYY,R(99)),(UXY,R(100)),(VX,R(101)),(VY,R(102)),(VXX,R(103)),    TRN03800
```

```
     8(VYY,R(104)),(VXY,R(105)),(AH,R(106)),(AK,R(107)),(UU,R(108)),     TRN03900
     9(VV,R(109)),(UU1,R(110)),(VV1,R(111))                              TRN04000
      EQUIVALENCE (HMI,R(112)),(HMJ,R(113)),(XP,R(114)),(YP,R(115)),     TRN04100
     1(S1,R(116)),(S2,R(117))                                            TRN04200
C                                                                        TRN04300
C     THIS SUBROUTINE CALCULATES THE TRAJECTORY OF THE CENTROID OF THE   TRN04400
C     CLOUD FROM THE RELEASE (SOURCE) POINT XSOURC,YSOURC. THE INCREMENTTRN04500
C     BETWEEN SUCCESSIVE POINTS IS 1/10 THE GRID SPACING (DX). ALSO,     TRN04600
C     THIS SUB CALCULATES THE LAYER WIND SPEED AND DEPTH AT EACH         TRN04700
C     TRAJECTORY POINT AND PASSES THIS INFO TO SUB DIFFUZ. THIS SUB      TRN04800
C     ASSUMES THE REAL TERRAIN GRID REGION IS A 31X31 MATRIX (WITH       TRN04900
C     EXPANDED REGION 41X41).                                            TRN05000
C     U,V,W - ARE THE U AND V COMPONENTS OF THE WIND FIELD SOLUTION      TRN05100
C             AND W IS THE LAYER HEIGHT (CONVERTED TO DEPTH) OF THE      TRN05200
C             WIND FIELD SOLUTION OVER THE REAL TERRAIN (31X31) MATRIX.  TRN05300
C             THESE DATA ARE TAKEN FROM THE ARRAYS MM,NN,PP.             TRN05400
C     MM,NN,PP - ARE THE U,V AND P COMPONENTS OF THE WIND FIELD SOLUTIONTRN05500
C                  TAKEN FROM THE FILE HOLDING THE LATEST WIND FIELD     TRN05600
C                  SOLUTION                                              TRN05700
C     THE LATEST SOLUTION FILE IS READ INTO COMMON DIFUZZ WHICH IS       TRN05800
C     GREATER THAN OR EQUAL IN SIZE TO COMMON MXBUFR DEFINED IN SUB      TRN05900
C     MXSTRT. THE INFO THAT IS NEEDED FROM THE FILE IS NETRAD (NET -     TRN06000
C     RADIATION INDEX) , XSOURC,YSOURC,ZSOURC (COORDINATES OF SOURCE FROTRN06100
C     ANALYSIS) AND THE M,N AND P ARRAYS FROM THE ANALYSIS PHASE. TO     TRN06200
C     SAVE STORAGE THE U,V,W AND MM,NN,PP ARRAYS SHARE THE SAME STORAGE  TRN06300
C     BLOCK IN COMMON DIFUZZ. WE MUST INPUT MM,NN AND PP AS 41X41 ARRAYSTRN06400
C     , BUT WE ARE ONLY GOING TO USE THE INTERNAL 31X31 PART OF THE      TRN06500
C     ARRAYS. IF THE PRESENT ARRAY SIZE IS CHANGED THE USER SHOULD       TRN06600
C     STUDY THE COMMON DIFUZZ AND THE ABOVE DIMENSION AND EQUIVALENCE    TRN06700
C     STATEMENTS TO MAKE SURE STORAGE IS ALLIGNED CORRECTLY..            TRN06800
C                                                                        TRN06900
C                                                                        TRN07000
      IHOURS = IHOUR*100+NINUTE                                          TRN07100
C     GET LAST WINDFIELD SOLUTION                                        TRN07200
      N2 = 3*IDP*JDP+68                                                  TRN07300
      K2 = 2                                                             TRN07400
      IF (KFLG1 .NE. 0) K2 = 1                                           TRN07500
      CALL BLKIN(N2,NETRAD,K2,MXLYRF,ISTS)                               TRN07600
```

```
      N2 = IDP-5                                                        TRN07700
      K2 = JDP-5                                                        TRN07800
C     PACK ARRAYS DOWN TO 31X31 MATRICES                                TRN07900
      DO 10 J=6,K2                                                      TRN08000
      DO 10 I=6,N2                                                      TRN08100
   10 W(N2-I+1,K2-J+1) = PP(IDP-I+1,JDP-J+1)                            TRN08200
      DO 20 J=6,K2                                                      TRN08300
      DO 20 I=6,N2                                                      TRN08400
   20 V(N2-I+1,K2-J+1) = NN(IDP-I+1,JDP-J+1)                            TRN08500
      DO 30 J=6,K2                                                      TRN08600
      DO 30 I=6,N2                                                      TRN08700
   30 U(N2-I+1,K2-J+1) = MM(IDP-I+1,JDP-J+1)                            TRN08800
C     GET TERRAIN ELEVATION HEIGHTS                                     TRN08900
      N1 = IDP*JDP                                                      TRN09000
      CALL BLKIN(N1,MM(1,1),1,WSTRRN,ISTS)                              TRN09100
C     CONVERT HEIGHT W TO DEPTH                                         TRN09200
      DO 40 J=6,K2                                                      TRN09300
      DO 40 I=6,N2                                                      TRN09400
   40 W(I-5,J-5) = W(I-5,J-5)-MM(I,J)                                   TRN09500
C     IF XSOUR AND YSOUR ARE INPUT USE THOSE                            TRN09600
      IF (XSOUR .GT. 0.0) XSOURC = XSOUR                                TRN09700
      IF (YSOUR .GT. 0.0) YSOURC = YSOUR                                TRN09800
      IF (ZSOUR .GT. 0.0) ZSOURC = ZSOUR                                TRN09900
C     CALCULATE X AND Y AXES                                            TRN10000
      IDP = IDP-10                                                      TRN10100
      JDP = JDP-10                                                      TRN10200
      X(1) = XSTRT                                                      TRN10300
      Y(1) = YSTRT                                                      TRN10400
      DO 50 I=2,IDP                                                     TRN10500
   50 X(I) = X(I-1)+XINC                                                TRN10600
      DO 60 J=2,JDP                                                     TRN10700
   60 Y(J) = Y(J-1)+YINC                                                TRN10800
      N2 = IDP-1                                                        TRN10900
      K2 = JDP-1                                                        TRN11000
      XMIN = AMIN1(XINC,YINC)                                           TRN11100
C     CALL MAXIMIM VECTOR                                               TRN11200
      UVMX = 0.0                                                        TRN11300
      DO 70 J=1,JDP                                                     TRN11400
```

```
      DO 70 I=1,IDP                                                     TRN11500
   70 UVMX = AMAX1(UVMX,U(I,J)*U(I,J)+V(I,J)*V(I,J))                    TRN11600
      UVMX = SQRT(UVMX)                                                 TRN11700
C     CALC INCREMENT IN METERS FOR TRAJECTORY POINTS                    TRN11800
      DX = 0.1*XMIN                                                     TRN11900
C     LAST U AND V COMP CALCULATED                                      TRN12000
      UUL = UVMX                                                        TRN12100
      VVL = UVMX                                                        TRN12200
C     SET CONSTANTS BECAUSE EVENLY SPACED GRID                          TRN12300
      HXYM = XMIN                                                       TRN12400
      HXM = 0.05*HXYM                                                   TRN12500
      HX = X(2)-X(1)                                                    TRN12600
      HY = Y(2)-Y(1)                                                    TRN12700
      HX1 = 0.5/HX                                                      TRN12800
      HY1 = 0.5/HY                                                      TRN12900
      HX5 = .5*HX                                                       TRN13000
      HY5 = .5*HY                                                       TRN13100
      HX21 = 2.0*HX1*HX1                                                TRN13200
      HY21 = 2.0*HY1*HY1                                                TRN13300
      HXY4 = HX1*HY1                                                    TRN13400
      U1 = 0.0                                                          TRN13500
      V1 = 0.0                                                          TRN13600
      U2 = 0.0                                                          TRN13700
      V2 = 0.0                                                          TRN13800
      XD = 0.0                                                          TRN13900
C     DELTA - TIME                                                      TRN14000
      DT = DX/(0.5*UVMX)                                                TRN14100
      T = 0.0                                                           TRN14200
      N1 = 0                                                            TRN14300
      K1 = 0                                                            TRN14400
      IB = 0                                                            TRN14500
      XO = XSOURC                                                       TRN14600
      YO = YSOURC                                                       TRN14700
      IF (XO .GT. X(NX).OR.XO .LT. X(1)) GO TO 200                      TRN14800
      IF (YO .GT. Y(NY).OR.YO .LT. Y(1)) GO TO 200                      TRN14900
C     LOOP THROUGH TRAJ POINTS UNTIL 310 OR LEAVES GRID.                TRN15000
      LINES = 80                                                        TRN15100
   80 CONTINUE                                                          TRN15200
```

```
      N = INT((XO-X(1))*HX1*2,0)+1
      IF (XO-X(N) .GT. HX5) N = N+1
      IF (N ,GE. NX) N = NX-1
      IF (N ,LE. 1) N = 2
      K = INT((YO-Y(1))*HY1*2,0)+1
      IF (YO-Y(K) .GT. HY5) K = K+1
      IF (K ,GE. NY) K=NY-1
      IF (K ,LE. 1) K = 2
C     CALC NEXT UU,VV
      NT = 0
      X1 = XO
      Y1 = YO
      IF (N1 .EQ. N.AND.K1 .EQ. K) GO TO 140
      N1 = N
      K1 = K
      UX = (U(N+1,K)-U(N-1,K))*HX1
      UY = (U(N,K+1)-U(N,K-1))*HY1
      UXX = HX21*(U(N-1,K)-U(N,K)-U(N,K)+U(N+1,K))
      UYY = HY21*(U(N,K-1)-U(N,K)-U(N,K)+U(N,K+1))
      UXY = HXY4*(U(N+1,K+1)-U(N-1,K+1)-U(N+1,K-1)+U(N-1,K-1))
      VX = (V(N+1,K)-V(N-1,K))*HX1
      VY = (V(N,K+1)-V(N,K-1))*HY1
      VXX = HX21*(V(N-1,K)-V(N,K)-V(N,K)+V(N+1,K))
      VYY = HY21*(V(N,K-1)-V(N,K)-V(N,K)+V(N,K+1))
      VXY = HXY4*(V(N+1,K+1)-V(N-1,K+1)-V(N+1,K-1)+V(N-1,K-1))
  140 CONTINUE
      AH = X1-X(N)
      AK = Y1-Y(K)
      UU = U(N,K)+AH*UX+AK*UY+AH*AH*UXX+AH*AK*UXY+AK*AK*UYY
      VV = V(N,K)+AH*VX+AK*VY+AH*AH*VXX+AH*AK*VXY+AK*AK*VYY
      IF (NT .NE. 0) GO TO 150
      NT = 3
      X1 = X1+UU*DT
      Y1 = Y1+VV*DT
      UU1 = UU
      VV1 = VV
      GO TO 140
  150 CONTINUE
```

```
      UU = 0.5*(UU+UU1)                                                 TRN19100
      VV = 0.5*(VV+VV1)                                                 TRN19200
  160 IF (IB .NE. 0) GO TO 170                                          TRN19300
      GO TO 175                                                         TRN19400
  165 HMI = HMJ                                                         TRN19500
      IB = IB+1                                                         TRN19600
  170 IF (UU) 173,171,173                                               TRN19700
  171 IF (VV) 173,172,173                                               TRN19800
  172 UU = UUL                                                          TRN19900
      VV = VVL                                                          TRN20000
  173 X1 = SQRT(UU*UU+VV*VV)                                            TRN20100
      UUL = UU                                                          TRN20200
      VVL = VV                                                          TRN20300
      DT = DX/X1                                                        TRN20400
      XO = XO+DT*UU                                                     TRN20500
      YO = YO+DT*VV                                                     TRN20600
      IF (XO.LT.X(1).OR.XO.GT.X(NX).OR.YO.LT.Y(1).OR.YO.GT.Y(NY)) GO TO TRN20700
     1200                                                               TRN20800
      T = T+DT                                                          TRN20900
      XD = XD+DX                                                        TRN21000
  175 IF (XO .GE. X(N2).AND.XO .LE. X(N2+1)) GO TO 177                  TRN21100
      N2 = NX-1                                                         TRN21200
  176 IF (XO .GE. X(N2)) GO TO 177                                      TRN21300
      N2 = N2-1                                                         TRN21400
      IF (N2 .GT. 1) GO TO 176                                          TRN21500
  177 IF (YO .GE. Y(K2).AND.YO .LE. Y(K2+1)) GO TO 179                  TRN21600
      K2 = NY-1                                                         TRN21700
  178 IF (YO .GE. Y(K2)) GO TO 179                                      TRN21800
      K2 = K2-1                                                         TRN21900
      IF (K2 .GT. 1) GO TO 178                                          TRN22000
C     CALC MIXING LAYER DEPTH                                           TRN22100
  179 XP = (XO-X(N2))/(X(N2+1)-X(N2))                                   TRN22200
      YP = (YO-Y(K2))/(Y(K2+1)-Y(K2))                                   TRN22300
      HMJ = (1.0-XP)*(1.0-YP)*W(N2,K2)+XP*(1.0-YP)*W(N2+1,K2)+          TRN22400
     1(1.0-XP)*YP*W(N2,K2+1)+XP*YP*W(N2+1,K2+1)                         TRN22500
      IF (IB .EQ. 0) GO TO 165                                          TRN22600
      HMI = 0.5*(HMJ+HMI)                                               TRN22700
      XX(IB) = XO                                                       TRN22800
```

```
      YY(IB) = YO
      UBAR(IB) = X1
      HM(IB) = HMI
      IF (KFLG2 .EQ. 0) GO TO 186
      LINES = LINES+1
      IF (LINES .LT. 57) GO TO 185
      WRITE (XOTFIL,9000) IULIAN,IMONTH,IDAY,IHOURS
      LINES = 8
  185 WRITE (XOTFIL,9001) IB,XX(IB),YY(IB),UU,VV,XD,UBAR(IB),DT,T,HM(IB)
  186 CONTINUE
      HMI = HMJ
C     DETERMINE IF CURVE IS MAKING A DECREASING SPIRAL IF YES STOP SOL.
      U1 = U1+UU
      V1 = V1+VV
      U2 = U2+ABS(UU)
      V2 = V2+ABS(VV)
      S1 = U1*U1+V1*V1
      S2 = U2*U2+V2*V2
      IF (S2/S1 .GE. 49.0) GO TO 200
  195 CONTINUE
      IB = IB+1
      IF (IB .LE. 310) GO TO 80
  200 CONTINUE
      IB = IB-1
 9000 FORMAT ('1',40X,'*-* SOLUTION FOR - JUL/M/D/H=',I3,'/',I2,'/',I2,'
     */',I4,'Z *-*'/8X,'*-* DIFFUSION MODEL SOURCE TRAJECTORY *-*'//  '
     1 POINT',5X,'X',11X,'Y',9X,'U',7X,'V',5X,'DISTANCE VELOCITY  DELTA-
     2T    TIME       HM'/' NUMBER  (METERS)     (METERS)  (M/SEC) (M/SEC)
     3  (METERS)  (M/SEC)   (SEC)     (SEC)    (METERS)'/1X,46('--'))
 9001 FORMAT (2X,I4,2F12.2,2F8.2,F10.2,2F9.2,F10.2,F9.2)
      RETURN
      END
```

```
      SUBROUTINE DIFFUZ                                                 DIF00100
      INCLUDE TIMES,LIST                                                DIF00200
      INCLUDE MESAGE,LIST                                               DIF00300
      INCLUDE FILES,LIST                                                DIF00400
      INCLUDE DIFUZZ,LIST                                               DIF00500
      INCLUDE COMWND,LIST                                               DIF00600
      INCLUDE SOURCE,LIST                                               DIF00700
      DATA NUMX/310/                                                    DIF00800
C **  SET LENGTH OF ARRAYS XX,YY,ETC IN NUMX                            DIF00900
C                                                                       DIF01000
C     THIS SUBROUTINE CALCULATES THE DIFFUSION PARAMETERS CONCENTRATION,DIF01100
C     DOSAGE, DEPOSITION, ETC AND OUTPUTS THE SOLUTION TO SAVE FILE     DIF01200
      DIMENSION SIGAT(7,5),SIGAM(7,5),SIGET(7,5),PWR(7,5),UBRT(5)       DIF01300
     1,SIGEN(7,5)                                                       DIF01400
      DIMENSION CHIT(310),DOST(310),SIGMY(310),YCHIT(310,3),YDOST(310,3)DIF01500
C     PASS NO. OF PNTS. IN TRAJECTORY THROUGH COMMON VIA HMIN           DIF01600
      EQUIVALENCE (HMIN,NPNT)                                           DIF01700
C     DEFINE STORAGE ADDRESSES FOR DIFFUSION CALCULATION ARRAYS         DIF01800
      EQUIVALENCE (CHIT,R(1)),(DOST,R(311)),(SIGMY,R(621)),             DIF01900
     1(YCHIT,R(931)),(YDOST,R(1861))                                    DIF02000
C                                                                       DIF02100
      EQUIVALENCE (NCALC,NETRAD)                                        DIF02200
      DIMENSION CONI(3),DOSI(3)                                         DIF02300
      EQUIVALENCE (KINDS,SURFTA),(CONI,POTMPR(1)),(DOSI,POTMPR(4))      DIF02400
      DIMENSION RLAPSE(3,4),TEMPSD(4,12)                                DIF02500
      DATA RLAPSE/.005,.003,.001,.015,.010,.005,.030,.020,.015,.040,.030DIF02600
     1,.020/                                                            DIF02700
      DATA TEMPSD/292.,286.,279.,272.,296.,290.,282.,275.,300.,294.,286.DIF02800
     1,277.,303.,299.,290.,282.,308.,304.,295.,287.,312.,308.,300.,292.,DIF02900
     2312.,308.,301.,294.,310.,307.,300.,293.,308.,304.,297.,289.,303., DIF03000
     3299.,291.,283.,297.,292.,284.,275.,293.,287.,280.,272./           DIF03100
      DATA SIGAT/26.,26.,21.,15.,11.,8.6,8.6,26.,22.,16.,11.,10.,8.6,8.6DIF03200
     1,19.,16.,13.,2*9.,2*7.8,14.,2*11.,8.,7.,2*6.7,2*9.,2*8.,7.,2*6.7/ DIF03300
      DATA SIGAM/2*-.109,-.133,-.112,-.120,2*-.189,-.109,-.128,-.123,   DIF03400
     1-.112,-.120,2*-.189,-.122,-.123,-.103,-.120,-.134,-.207,-.189,    DIF03500
     2-.105,2*-.117,-.148,-.126,-.287,-.189,-.1,-.117,-.11,-.146,-.126, DIF03600
     3-.287,-.189/                                                      DIF03700
      DATA SIGEN/2*.1,.07,2*0.,-.20,-.3,.1,.13,.07,2*0.,-.2,-.25,.15,   DIF03800
```

```
     1.15,.13,2*0.,2*-.2,2*.2,.15,2*0.,2*-.15,.25,2*.2,2*0.,2*-.15/    DIF03900
      DATA SIGET/2*8.7,7.0,5.0,3.7,2*2.9,8.7,7.4,5.3,3.7,3.3,2*2.9,6.3, DIF04000
     15.3,4.3,2*3.0,2*2.6,4.7,2*3.7,2*2.7,2*2.2,2*3.0,3*2.7,2*2.2/      DIF04100
      DATA PWR/6*.20,.30,.20,.17,4*.2,.25,2*.15,2*.17,3*.2,2*.10,5*.15, DIF04200
     1.05,3*.10,3*.15/                                                  DIF04300
      DATA UBRT/1.0,3.0,5.0,7.0,99.0/,RAD/.01745329/                    DIF04400
      PTERP(A,B,C,D,E) = A-(B-C)*(A-D)/(B-E)                            DIF04500
      CISO(A,B) = SQRT(A*ALOG(B))                                       DIF04600
C                                                                       DIF04700
      CHIL = 1.0E20                                                     DIF04800
      IF (KINDSO .EQ. 0) KFLG = 1                                       DIF04900
      SUMN = 0.0                                                        DIF05000
      ALPHAI = 1.0/ALPHA                                                DIF05100
      BETAI = 1.0/BETA                                                  DIF05200
      LFLG = 0                                                          DIF05300
      MFLG = 0                                                          DIF05400
      IF (KINDSO .NE. 0) GO TO 10                                       DIF05500
      QP = Q*6.3493636E-2                                               DIF05600
      HP = H                                                            DIF05700
      HPP = H                                                           DIF05800
      IF(HP.LT. 5.0) HP = 5.0                                           DIF05900
      MFLG = 1                                                          DIF06000
      GO TO 20                                                          DIF06100
   10 QP = Q*.159154943                                                 DIF06200
      IF (TEMPSF .GT. 0.0) SURFTA = TEMPSF                              DIF06300
      IF (SURFTA .GT. 0.0) GO TO 20                                     DIF06400
C     DEFAULT SURFACE TEMP                                              DIF06500
      IF (NETRAD .EQ. 4) GO TO 17                                       DIF06600
      IF (NETRAD .GE. 2) GO TO 16                                       DIF06700
      IF (NETRAD .GE. 0) GO TO 15                                       DIF06800
      L1 = 4                                                            DIF06900
      GO TO 18                                                          DIF07000
   15 L1 = 3                                                            DIF07100
      GO TO 18                                                          DIF07200
   16 L1 = 2                                                            DIF07300
      GO TO 18                                                          DIF07400
   17 L1 = 1                                                            DIF07500
   18 SURFTA = TEMPSD(L1,MONTH)                                         DIF07600
```

```
   20 CONTINUE                                                          DIF07700
      LINE = 80                                                         DIF07800
C     BEGIN LOOP OVER TRAJECTORY POINTS                                 DIF07900
      SIGXL = SIGXO                                                     DIF08000
      SIGZL = SIGZO                                                     DIF08100
      SIGYL = SIGYO                                                     DIF08200
      XO = XSOURC                                                       DIF08300
      YO = YSOURC                                                       DIF08400
      DO 500 K=1,NPNT                                                   DIF08500
      DX = SQRT((XX(K)-XO)**2+(YY(K)-YO)**2)                            DIF08600
      XO = XX(K)                                                        DIF08700
      YO = YY(K)                                                        DIF08800
      CHI = 0.0                                                         DIF08900
      DOS = 0.0                                                         DIF09000
C     CALCULATE WIND SPEED POWER LAW EXPONENT CATEGORY                  DIF09100
      DO 50 J=1,5                                                       DIF09200
      IF (UBAR(K) .LT. UBRT(J)) GO TO 51                                DIF09300
   50 CONTINUE                                                          DIF09400
      J = 5                                                             DIF09500
   51 CONTINUE                                                          DIF09600
      I = 5-NETRAD                                                      DIF09700
      PWRS = PWR(I,J)                                                   DIF09800
C     CALCULATE WIND SPEED AT 5 METERS                                  DIF09900
      UBAR5 = UBAR(K)*(HM(K)-5.0)*5.0**PWR(I,J)*(1.0+PWR(I,J))/         DIF10000
     1(HM(K)**(1.0+PWR(I,J))-5.0**(1.0+PWR(I,J)))                       DIF10100
C     IF INSTANTANEOUS OR CONTINUOUS SOURCE                             DIF10200
      IF (JFLG .NE. 0 .OR. KINDSO .EQ. 0) GO TO 85                      DIF10300
      JFLG = 1                                                          DIF10400
      UBARH = UBAR5*(H*0.2)**PWRS                                       DIF10500
      F = 1.0                                                           DIF10600
      IF (UBARH .LE. EXTVEL*6.6666666E-1) GO TO 61                      DIF10700
      IF (UBARH .GT. EXTVEL) GO TO 60                                   DIF10800
      F = (3.0*EXTVEL-3.0*UBARH)/EXTVEL                                 DIF10900
      GO TO 61                                                          DIF11000
   60 F = 0.0                                                           DIF11100
   61 IF (NETRAD .GE. 2.OR.UBARH .GE. 5.0) GO TO 83                     DIF11200
C     CALC LAPSE RATE OF POTENTIAL TEMP                                 DIF11300
      IF (RTLAPS .GT. 0.0) GO TO 81                                     DIF11400
```

```
      B = 0.0                                                           DIF11500
      L = 0                                                             DIF11600
   62 L = L + 1                                                         DIF11700
      IF (POTMPR(L) .LE. 0.0) GO TO 65                                  DIF11800
      IF (ZELEVR(L) .LE. ZSOURC+H) GO TO 62                             DIF11900
      A1 = PTERP(POTMPR(L),ZELEVR(L),ZSOURC+H,POTMPR(L-1),ZELEVR(L-1))  DIF12000
      B = ZELEVR(L)-ZSOURC-H                                            DIF12100
      A = (POTMPR(L)-A1)*B                                              DIF12200
   63 L = L+1                                                           DIF12300
      IF (L .GT. 25) GO TO 65                                           DIF12400
      IF (POTMPR(L) .LE. 0.0) GO TO 65                                  DIF12500
      IF (ZELEVR(L) .GT. ZSOURC+H+200.0) GO TO 64                       DIF12600
      A1 = ZELEVR(L)-ZELEVR(L-1)                                        DIF12700
      A = A+(POTMPR(L)-POTMPR(L-1))*A1                                  DIF12800
      B = B+A1                                                          DIF12900
      GO TO 63                                                          DIF13000
   64 A1 = PTERP(POTMPR(L),ZELEVR(L),ZSOURC+H+200.0,POTMPR(L-1),ZELEVR(LDIF13100
     1-1))                                                              DIF13200
      A2 = ZSOURC+H+200.0-ZELEVR(L-1)                                   DIF13300
      A = A+(A1-POTMPR(L-1))*A2                                         DIF13400
      B = B+A2                                                          DIF13500
   65 IF (B .GT. 0.0) GO TO 80                                          DIF13600
C     DEFAULT LAPSE RATE                                                DIF13700
      IF (UBARH .LE. 1.4) GO TO 67                                      DIF13800
      IF (UBARH .LE. 2.9) GO TO 66                                      DIF13900
      L1 = 3                                                            DIF14000
      GO TO 68                                                          DIF14100
   66 L1 = 2                                                            DIF14200
      GO TO 68                                                          DIF14300
   67 L1 = 1                                                            DIF14400
   68 IF (NETRAD .GE. 1) GO TO 71                                       DIF14500
      IF (NETRAD .EQ. 0) GO TO 70                                       DIF14600
      IF (NETRAD .EQ. -1) GO TO 69                                      DIF14700
      L2 = 4                                                            DIF14800
      GO TO 72                                                          DIF14900
   69 L2 = 3                                                            DIF15000
      GO TO 72                                                          DIF15100
   70 L2 = 2                                                            DIF15200
```

```
      GO TO 72                                                          DIF15300
   71 L2 = 1                                                            DIF15400
   72 RTLAPS = RLAPSE(L1,L2)                                            DIF15500
      GO TO 81                                                          DIF15600
   80 RTLAPS = A/B                                                      DIF15700
   81 IF (RTLAPS .LE. 0.0) GO TO 71                                     DIF15800
      A2 = EXTVEL*RADIUS*RADIUS*SURFTA/(UBARH*RTLAPS)                   DIF15900
      A4 = 9.8*RTLAPS/SURFTA                                            DIF16000
      A1 = 0.1*UBARH/SQRT(A4)                                           DIF16100
      A3 = H/A1                                                         DIF16200
      A1 = 3.1415926*A1                                                 DIF16300
      IF (A1 .GE. H) GO TO 82                                           DIF16400
      HP =    2.397*F*(A2*(1.0-SURFTA/STTEMP))**.3333333                DIF16500
      GAMT = .66                                                        DIF16600
      GO TO 84                                                          DIF16700
   82 HP =    1.903*F*(A2*(1.0-COS(A3)))**.3333333                      DIF16800
      GAMT = .66                                                        DIF16900
      GO TO 84                                                          DIF17000
   83 HP =    F*((15.98/UBARH)*(RADIUS*RADIUS*EXTVEL*(1.0-SURFTA/STTEMP) DIF17100
     1*H*H)**.3333333)                                                  DIF17200
      GAMT = .6                                                         DIF17300
   84 SIGYL = (GAMT*HP+RADIUS)*.465116279                               DIF17400
      SIGXL = SIGYL                                                     DIF17500
      SIGZL = SIGYL                                                     DIF17600
      HP = HP+H                                                         DIF17700
      IF (HP .LT. 5.0) HP = 5.0                                         DIF17800
      HPP = HP                                                          DIF17900
   85 IF (K .GT. 1) GO TO 86                                            DIF18000
      IF (HPP .LE. HM(1)) GO TO 86                                      DIF18100
      WRITE (XOTFIL,1005)                                               DIF18200
 1005 FORMAT   ('0 SOURCE RISE HEIGHT IS ABOVE HM NO CALCULATIONS PRODUCDIF18300
     1ED, OUTPUT HAS ZEROES'/)                                          DIF18400
      TYPE3 = -1                                                        DIF18500
      GO TO 501                                                         DIF18600
   86 IF (LFLG .EQ. 1) GO TO 87                                         DIF18700
      IF (HPP .LE. HM(K)) GO TO 88                                      DIF18800
      LFLG = 1                                                          DIF18900
   87 HPP = HM(K)                                                       DIF19000
```

```
   88 CONTINUE
C     CALCULATE WIND SPEED CATEGORY
      DO 100 J=1,5
      IF (UBAR5 .LT. UBRT(J)) GO TO 101
  100 CONTINUE
      J = 5
  101 CONTINUE
C     CALCULATE SIGMA-A
      SIGAP = SIGAT(I,J)*RAD*(HP*.2)**SIGAM(I,J)
      IF (KINDSO .EQ. 0) SIGAP = SIGAP*(TAU*1.6666666E-3)**.2
C     CALCULATE SIGMA-E
      SIGEP = SIGET(I,J)*RAD*(HP*.2)**SIGEN(I,J)
C     CALCULATE SIGMA-Y
      SIGY = SIGAP*(DX+(SIGYL/SIGAP)**ALPHAI)**ALPHA
C     CALCULATE SIGMA-Z
      SIGZ = SIGEP*(DX+(SIGZL/SIGEP)**BETAI)**BETA
C     CALCULATE THE TOP AND BOTTOM OF THE CLOUD
      ZT = AMIN1(H+2.15*SIGZ,HM(K))
      ZB = AMAX1(H-2.15*SIGZ,0.0)
C     CALCULATE THE MEAN WIND SPEED IN THE LAYER CONTAINING THE CLOUD
      UBARN = UBAR5*(ZT**(1.0+PWRS)-ZB**(1.0+PWRS))/((1.0+PWRS)*5.0**PWR
     1S*(ZT-ZB))
C     CALCULATE THE WIND SPEED SHEAR
      DELTAU = UBAR5*(ZT**PWRS-ZB**PWRS)/5.0**PWRS
C     CALCULATE SIGMA-X
      SIGX = 1.3953488E-1*DELTAU*DX/UBARN +SIGXL
C     CALCULATE THE DEPLETION TERMS
      SUMN = SUMN+DX/UBARN
      DECAYS = 1.0
      IF (DECAY .LE. 0.0) GO TO 120
      DECAYS = EXP(-DECAY*SUMN)
  120 CONTINUE
      TLAMDA = 1.0
      IF (BLAMDA .LE. 0.0) GO TO 130
      TLAMDA = EXP(-BLAMDA*SUMN)
  130 CONTINUE
C     CALCULATE THE VERTICAL TERM
      SIGZI = -0.5/(SIGZ*SIGZ)
```

```
      VERT = 0.0                                                        DIF22900
      IF (SIGZI .GE. 0.0) GO TO 220                                     DIF23000
      AR2 = VS*SUMN                                                     DIF23100
      AR1 = -HPP+Z+AR2                                                  DIF23200
      AR2 = HPP+Z-AR2                                                   DIF23300
      C = AR2*AR2*SIGZI                                                 DIF23400
      IF (C .LT. -30.0) GO TO 220                                       DIF23500
      D = AR1*AR1*SIGZI                                                 DIF23600
      IF (D .LT. -30.0) GO TO 200                                       DIF23700
      VERT = EXP(D)                                                     DIF23800
  200 VERT = VERT+GAMMA*EXP(C)                                          DIF23900
      C = 1                                                             DIF24000
      D = GAMMA                                                         DIF24100
      E = D*D                                                           DIF24200
      A = 0.0                                                           DIF24300
  210 A = A+2.0                                                         DIF24400
      B = A*HM(K)                                                       DIF24500
      A1 = B-AR2                                                        DIF24600
      A1 = A1*A1*SIGZI                                                  DIF24700
      IF (A .LE. 2.0.AND.A1 .GT. -30.0) GO TO 215                       DIF24800
      IF (A1 .LT. -10.0) GO TO 220                                      DIF24900
  215 CONTINUE                                                          DIF25000
      A2 = B+AR1                                                        DIF25100
      A3 = B-AR1                                                        DIF25200
      A4 = B+AR2                                                        DIF25300
      VERT = VERT +C*EXP(A1)+D*(EXP(A2*A2*SIGZI)+EXP(A3*A3*SIGZI))+     DIF25400
     1E*EXP(A4*A4*SIGZI)                                                DIF25500
      C = D                                                             DIF25600
      D = E                                                             DIF25700
      E = E*GAMMA                                                       DIF25800
      GO TO 210                                                         DIF25900
  220 CONTINUE                                                          DIF26000
      CHI = (QP/(SIGZ*SIGY))*VERT*DECAYS*TLAMDA                         DIF26100
      IF (KINDSO .NE. 0) GO TO 230                                      DIF26200
C     CALCULATE CONCENTRATION EXCEPT FOR LATERAL TERM FOR INSTANTANEOUS DIF26300
C     VOLUME SOURCE                                                     DIF26400
      CHI = CHI/SIGX                                                    DIF26500
      GO TO 240                                                         DIF26600
```

```
C     CALCULATE CONCENTRATION EXCEPT FOR LATERAL TERM FOR CONTINUOUS    DIF26700
C     VOLUME SOURCE                                                     DIF26800
  230 CHI = CHI/UBARN                                                   DIF26900
  240 CONTINUE                                                          DIF27000
      IF (K .EQ.1) GO TO 300                                            DIF27100
      IF (MFLG .EQ. 0.AND.CHI .LT. CHIL) MFLG = 1                       DIF27200
      IF (CHI .LE. CHIL) GO TO 300                                      DIF27300
      IF (MFLG .EQ. 0) GO TO 300                                        DIF27400
C     CON IS GREATER     - CHANGE SIGX AND SIGY                         DIF27500
      A1 = SQRT(CHI/CHIL)                                               DIF27600
      SIGX = A1*SIGX                                                    DIF27700
      SIGY = A1*SIGY                                                    DIF27800
      CHI = CHIL                                                        DIF27900
  300 CONTINUE                                                          DIF28000
      CHIL = CHI                                                        DIF28100
      SIGXL = SIGX                                                      DIF28200
      SIGYL = SIGY                                                      DIF28300
      SIGZL = SIGZ                                                      DIF28400
      IF (KINDSO .NE. 0) GO TO 310                                      DIF28500
C     CALCULATE WIND SPEED AT THE CALCULATION HEIGHT Z                  DIF28600
      A1 = 0.2                                                          DIF28700
      IF (Z .GT. 1.0) A1 = A1*Z                                         DIF28800
      UBARZ = UBAR5*(A1)**PWRS                                          DIF28900
C     CALCULATE THE DOSAGE FOR AN INSTANTANEOUS SOURCE                  DIF29000
      DOS = CHI*2.50662827*SIGX/UBARZ                                   DIF29100
C     SAVE CALCULATED VALUES                                            DIF29200
  310 CONTINUE                                                          DIF29300
      CHIT(K) = CHI                                                     DIF29400
      DOST(K) = DOS                                                     DIF29500
      SIGMY(K) = SIGY                                                   DIF29600
C     CALCULATE ISOPLETHS OF CON, DOS AND/OR DEP                        DIF29700
      N = 0                                                             DIF29800
      SIGY = 2.0*SIGY*SIGY                                              DIF29900
  320 N = N+1                                                           DIF30000
      IF (CONISS(N) .LE. 0.0) GO TO 330                                 DIF30100
      A = CHI/CONISS(N)                                                 DIF30200
      IF (A .LE. 1.0) GO TO 330                                         DIF30300
      YCHIT(K,N) = CISO(SIGY,A)                                         DIF30400
```

```
      GO TO 331                                                         DIF30500
  330 YCHIT(K,N) = 0.0                                                  DIF30600
  331 IF (DOSISS(N) .LE. 0.0) GO TO 340                                 DIF30700
      A = DOS/DOSISS(N)                                                 DIF30800
      IF (A .LE. 1.0) GO TO 340                                         DIF30900
      YDOST(K,N) = CISO(SIGY,A)                                         DIF31000
      GO TO 341                                                         DIF31100
  340 YDOST(K,N) = 0.0                                                  DIF31200
  341 IF (N .LT. 3) GO TO 320                                           DIF31300
C     OUTPUT RESULTS                                                    DIF31400
      LINE = LINE+1                                                     DIF31500
      IF (KINDSO .EQ. 0) LINE = LINE +1                                 DIF31600
      IF (LINE .LT. 57) GO TO 480                                       DIF31700
      IHOURS = IHOUR*100+MINUTE                                         DIF31800
      WRITE (XOTFIL,999) IULIAN,IMONTH,IDAY,IHOURS                      DIF31900
  999 FORMAT ('1',40X,'*-* SOLUTION FOR - JUL/M/D/H=',I3,'/',I2,'/',I2,'DIF32000
     1/',I4,'Z *-*')                                                    DIF32100
      WRITE (XOTFIL,1000) CONISS                                        DIF32200
 1000 FORMAT (' ',17X,'*-* TRAJECTORY *-*',25X,'/',10X,'*-* ISOPLETH HALDIF32300
     1FWIDTHS *-*'/' CENTROID LOCATION    SIGMA-Y (M)  CONCENTRATION (UNDIF32400
     2ITS/M**3)',3('  CHI=',E11.6)/5X,'X (M)',5X,'Y (M)')               DIF32500
      IF (KINDSO .EQ. 0) WRITE (XOTFIL,1001) DOSISS                     DIF32600
 1001 FORMAT (35X,'DOSAGE (UNIT SECONDS/M**3)',3('  DOS=',E11.6))       DIF32700
      WRITE (XOTFIL,1002)                                               DIF32800
 1002 FORMAT (1X,56('--'))                                              DIF32900
      LINE = 8                                                          DIF33000
      IF (KINDSO .EQ. 0) LINE = LINE + 2                                DIF33100
  480 WRITE (XOTFIL,1003) XX(K),YY(K),SIGMY(K),CHIT(K),(YCHIT(K,L),L=1,3DIF33200
     1)                                                                 DIF33300
      IF (KINDSO .EQ. 0) WRITE (XOTFIL,1004) DOST(K),(YDOST(K,L),L=1,3) DIF33400
 1003 FORMAT (1X,F8.1,F10.1,F12.2,9X,E13.8,7X,3(7X,F10.2))              DIF33500
 1004 FORMAT (40X,E13.8,7X,3(7X,F10.2))                                 DIF33600
  500 CONTINUE                                                          DIF33700
      NCALC = K                                                         DIF33800
  501 CONTINUE                                                          DIF33900
C     OUTPUT SOLUTIONS TO SAVE FILE                                     DIF34000
      KINDS = KINDSO                                                    DIF34100
      DO 550 I=1,3                                                      DIF34200
```

```
      CONI(I) = CONISS(I)                                               DIF34300
  550 DOSI(I) = DOSISS(I)                                               DIF34400
      CALL BLKOUT(4090,NCALC,1,TRDFFL,ISTS)                             DIF34500
      RETURN                                                            DIF34600
      END                                                               DIF34700
```

```
      SUBROUTINE PLTPRG                                                 PLT00100
      INCLUDE MESAGE,LIST                                               PLT00200
      INCLUDE BUFRMX,LIST                                               PLT00300
      INCLUDE FILES,LIST                                                PLT00400
      INCLUDE CONDWN,LIST                                               PLT00500
      INCLUDE WNDDTB,LIST                                               PLT00600
      INCLUDE N1,LIST                                                   PLT00700
      DIMENSION COND(310,3,2),DCONC(3,2)                                PLT00800
      DIMENSION IMTL(4),IMTM(4),IMTN(3)                                 PLT00900
      DIMENSION CNISS(3),DSISS(3),CDONC(3,2)                            PLT01000
      COMMON /DTA1/J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLL PLT01100
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     PLT01200
      DIMENSION XX(310),YY(310),SIGMY(310),CHIT(310),DOST(310),         PLT01300
     1YCHIT(310,3),YDOST(310,3),DOSI(3),CONI(3)                         PLT01400
      DIMENSION PLTM(200),SPACE(1),VAR(200,2),VOL(200,2),AREA(1),ITL(8),PLT01500
     1UM(1),X(31),Y(31),H(41,41),OUTS(1000),OUTR(3410),IDLBL(4,2)       PLT01600
      EQUIVALENCE (SPACE,P),(AREA,H),(UM,M),(OUTS,M),(TIM,SURFTA),(KINDSPLT01700
     10,SURFTA),(CONI,POTMPR(1)),(DOSI,POTMPR(4)),(NN,NCNTO)            PLT01800
      EQUIVALENCE (YCHIT,COND),(CONI,DCONC)                             PLT01900
      EQUIVALENCE (CNISS(1),CDONC(1,1)),(DSISS(1),CDONC(1,2))           PLT02000
      EQUIVALENCE (M,PLTM),(OUTS(201),VAR),(OUTS(601),VOL),(OUTR,UO),   PLT02100
     1(XX,OUTR(1)),(YY,OUTR(311)),(SIGMY,OUTR(1861)),(CHIT,OUTR(1241)), PLT02200
     2(DOST,OUTR(1551)),(YCHIT,OUTR(2171)),(YDOST,OUTR(3101))           PLT02300
      DATA IMTL/'INSTANTANEOUS SOURCE'/                                 PLT02400
      DATA IMTM/'CONTINUOUS SOURCE   '/                                 PLT02500
      DATA IMTN/'COPPER','LEAD',' '/                                    PLT02600
      DATA RAD/.01745/                                                  PLT02700
      DATA ITL/'TERRAIN HT.','  (METERS)',' VORTICITY',2*' '/           PLT02800
      DATA SCL/1.0E-4/                                                  PLT02900
      DATA IEVEN/1/                                                     PLT03000
      DATA ISW/0,1,18*0/                                                PLT03100
      DATA CONT/1219.2,1524.,1828.8,2133.6,2438.4,2743.2,3048.,3352.8/  PLT03200
      DATA NCONT/8/,NCONTS/16/,CONTS/50.,100.,200.,300.,400.,600.,800., PLT03300
     11000.,1200.,1400.,1600.,1800.,2000.,2200.,2400.,2600./            PLT03400
      DATA NCRV/1/                                                      PLT03500
      DATA X/300000.,305000.,310000.,315000.,320000.,325000.,330000.,   PLT03600
     1335000.,340000.,345000.,350000.,355000.,360000.,365000.,370000.,  PLT03700
     2375000.,380000.,385000.,390000.,395000.,400000.,405000.,410000.,  PLT03800
```

```
      3415000.,420000.,425000.,430000.,435000.,440000.,445000.,450000./  PLT03900
       DATA Y/3520000.,3525000.,3530000.,3535000.,3540000.,3545000.,      PLT04000
      13550000.,3555000.,3560000.,3565000.,3570000.,3575000.,3580000.,    PLT04100
      23585000.,3590000.,3595000.,3600000.,3605000.,3610000.,3615000.,    PLT04200
      33620000.,3625000.,3630000.,3635000.,3640000.,3645000.,3650000.,    PLT04300
      43655000.,3660000.,3665000.,3670000./                               PLT04400
       DATA XST/5*0.0/,YST/5*0.0/                                         PLT04500
       DATA RUNID/10*' '/                                                 PLT04600
       DATA IDLBL/'CONCENTRATION ISOPLETHS','DOSAGE ISOPLETHS',' '/       PLT04700
       CISO(A,B) = SQRT(A*ALOG(B))                                        PLT04800
       NAMELIST /QLST3/ ISW,XST,YST,CONT,NCONT,CONTS,NCONTS,              PLT04900
      1SCL,CNISS,DSISS                                                    PLT05000
       IF (TYPE3 .EQ. 0) GO TO 1020                                       PLT05100
       READ (XINFIL,QLST3)                                                PLT05200
       DO 1000 NCRV = 1,5                                                 PLT05300
       IF (XST(NCRV) .LE. 0.0) GO TO 1010                                 PLT05400
 1000  CONTINUE                                                           PLT05500
       NCRV = 6                                                           PLT05600
 1010  NCRV = NCRV-1                                                      PLT05700
 1020  IF (NCRV .GT. 0) GO TO 1030                                        PLT05800
       NCRV = 1                                                           PLT05900
 1030  CONTINUE                                                           PLT06000
       IF (ISW(11) .LT. 0) NCRV = -NCRV                                   PLT06100
       IF (ISW(11) .EQ. 0) NCRV = 0                                       PLT06200
       IOF = 0                                                            PLT06300
       IST = 6                                                            PLT06400
       JST = 6                                                            PLT06500
       IND = IDP-5                                                        PLT06600
       JND = JDP-5                                                        PLT06700
       NX = IND-JST+1                                                     PLT06800
       NY = JND-JST+1                                                     PLT06900
       INTO = IDP*JDP                                                     PLT07000
       IDIM = IDP                                                         PLT07100
       JDIM = JDP                                                         PLT07200
       NPLOT = ISW(3)+ISW(5)+ISW(8)+IABS(ISW(9))+IABS(ISW(10))+IABS(ISW(1PLT07300
      1))                                                                 PLT07400
       IPLOT = NPLOT+ISW(2)                                               PLT07500
       IF (NPLOT .EQ. 0) GO TO 1900                                       PLT07600
```

```
 1300 CONTINUE                                                            PLT07700
      JNTO = INTO*3+68                                                    PLT07800
      LSW = 0                                                             PLT07900
 1400 CONTINUE                                                            PLT08000
      IOF = IOF+1                                                         PLT08100
      CALL BLKIN(JNTO,NETRAD,IOF,SPARE,ISTS)                              PLT08200
      IF (NETRAD .LT. 0) GO TO 1700                                       PLT08300
      CALL BLKIN(INTO,H,1,WSTRRN,ISTS)                                    PLT08400
      E1 = TIM*1.6666666E-2                                               PLT08500
      IHOURS = IHOUR*100+NINUTE                                           PLT08600
      ENCODE (60,2001,RUNID,J) IULIAN,IMONTH,IDAY,IHOURS,NCNTO,E1         PLT08700
      MFLGS = 0                                                           PLT08800
      IF (XST(1) .LE. 0.0) XST(1) = XSOURC                                PLT08900
      IF (YST(1) .LE. 0.0) YST(1) = YSOURC                                PLT09000
      IF (ISW(1) .EQ. 0) GO TO 1520                                       PLT09100
      IF (ISW(1) .LT. 0.AND.LSW .EQ. 0) GO TO 1520                        PLT09200
      IF (ISW(1) .GT. 0.AND.LSW.EQ. 1) GO TO 1520                         PLT09300
      E9 = TIM*.27777777E-3                                               PLT09400
      DO 1510 J=JST,JND                                                   PLT09500
      LINES = 60                                                          PLT09600
      DO 1510 I=IST,IND                                                   PLT09700
      E1 = 1.0/P(I,J)                                                     PLT09800
      E2 = M(I,J)*E1                                                      PLT09900
      E3 = N(I,J)*E1                                                      PLT10000
      E1 = P(I,J)+H(I,J)                                                  PLT10100
 1485 E4 = H(I,J)                                                         PLT10200
      J3 = 1                                                              PLT10300
 1490 LINES = LINES+1                                                     PLT10400
      IF (LINES .LT. 57) GO TO 1501                                       PLT10500
      LINES = 7                                                           PLT10600
      WRITE (XOTFIL,2002) IULIAN,IMONTH,IDAY,IHOURS,ITL(J3),ITL(J3+1),   PLT10700
     1NN,E9                                                               PLT10800
      J4 = J3+3                                                           PLT10900
      IF (IABS(ISW(1)).EQ. 2) GO TO 1500                                  PLT11000
      WRITE (XOTFIL,2003) (ITL(J5),J5=J3,J4)                              PLT11100
      GO TO 1502                                                          PLT11200
 1500 WRITE (XOTFIL,2004) (ITL(J5),J5=J3,J4)                              PLT11300
 1501 IF (IABS(ISW(1)).EQ. 1) GO TO 1502                                  PLT11400
```

```
      CALL HEC003(E2,E3)                                                PLT11500
 1502 J4 = I-IST+1                                                      PLT11600
      J5 = J-JST+1                                                      PLT11700
      WRITE (XOTFIL,2005) J4,J5,X(J4),Y(J5),E2,E3,E1,E4                 PLT11800
 1510 CONTINUE                                                          PLT11900
      LINES = 60                                                        PLT12000
 1520 CONTINUE                                                          PLT12100
      IF (ISW(8) .EQ. 0) GO TO 1600                                     PLT12200
C     PLOT TERRAIN CONTOURS                                             PLT12300
      ISW(8) = 0                                                        PLT12400
      CALL HEC004(X(1),X(NX),Y(1),Y(NY),1)                              PLT12500
      CALL HEC005(X(1),Y(1),M,H,P,0,N)                                  PLT12600
      CALL HEC015(0,0,0)                                                PLT12700
 1600 IF (ISW(9) .EQ. 0) GO TO 1620                                     PLT12800
      IF (ISW(9) .LT. 0.AND.LSW .EQ. 0) GO TO 1620                      PLT12900
      IF (ISW(9) .GT. 0.AND.LSW .EQ. 1) GO TO 1620                      PLT13000
C     PLOT TERRAIN CONTOURS FOR VECTORS                                 PLT13100
      CALL HEC004(X(1),X(NX),Y(1),Y(NY),2)                              PLT13200
      IF (IABS(ISW(9)) .EQ. 2) GO TO 1610                               PLT13300
      CALL HEC005(X(1),Y(1),M,H,P,0,N)                                  PLT13400
      MFLGS = MFLGS+1                                                   PLT13500
 1610 CONTINUE                                                          PLT13600
C     PLOT VECTORS AND TRAJECTORIES                                     PLT13700
      CALL BLKIN(JNTO,NETRAD,IOF,SPARE,ISTS)                            PLT13800
      J2 = 0                                                            PLT13900
      DO 1615 J=JST,JND                                                 PLT14000
      J2 = J2+1                                                         PLT14100
      J1 = (J2-1)*NX                                                    PLT14200
      DO 1615 I=IST,IND                                                 PLT14300
      J1 = J1+1                                                         PLT14400
      E1 = 1.0/P(I,J)                                                   PLT14500
      UM(J1) = M(I,J)*E1                                                PLT14600
      SPACE(J1) = P(I,J)                                                PLT14700
      AREA(J1) = N(I,J)*E1                                              PLT14800
 1615 CONTINUE                                                          PLT14900
      CALL HEC006(X(1),Y(1),UM,AREA,NX,NY,SPACE,XMIN)                   PLT15000
      MFLGS = MFLGS+1                                                   PLT15100
      IF (IABS(ISW(10)) .EQ. 1) GO TO 1620                              PLT15200
```

```
      CALL HEC015(0,XM)                                                 PLT15300
 1620 IF (ISW(10) .EQ. 0) GO TO 1640                                    PLT15400
      IF (IABS(ISW(10)) .EQ. 2) GO TO 1624                              PLT15500
C     ISW(10) = 1 OR -1                                                 PLT15600
      IF (ISW(9) .LT. 0.AND.LSW .EQ. 0) GO TO 1640                      PLT15700
      IF (ISW(9) .GT. 0.AND.LSW .EQ. 1) GO TO 1640                      PLT15800
      GO TO 1625                                                        PLT15900
 1624 IF (ISW(10) .LT. 0.AND.LSW .EQ. 0) GO TO 1640                     PLT16000
      IF (ISW(10) .GT. 0.AND.LSW .EQ. 1) GO TO 1640                     PLT16100
 1625 CONTINUE                                                          PLT16200
C     PLOT LAYER DEPTH OR HEIGHT                                        PLT16300
      CALL BLKIN(INTO,H,1,WSTRRN,ISTS)                                  PLT16400
      CALL BLKIN(JNTO,NETRAD,IOF,SPARE,ISTS)                            PLT16500
      IF (IABS(ISW(10)) .EQ. 1) GO TO 1630                              PLT16600
      CALL HEC004(X(1),X(NX),Y(1),Y(NY),3)                              PLT16700
      IF (IABS(ISW(10)) .NE. 2) GO TO 1630                              PLT16800
      CALL HEC005(X(1),Y(1),M,H,P,0,N)                                  PLT16900
      MFLGS = MFLGS+1                                                   PLT17000
 1630 CALL HEC005(X(1),Y(1),M,H,P,1,N)                                  PLT17100
      CALL HEC015(0,XM)                                                 PLT17200
 1640 CONTINUE                                                          PLT17300
      GO TO 1400                                                        PLT17400
 1700 CONTINUE                                                          PLT17500
      IF (LSW .EQ. 1) GO TO 1710                                        PLT17600
      LSW = 1                                                           PLT17700
      IOF = IOF-2                                                       PLT17800
      IF (ISW(1) .LT. 0.OR.ISW(9) .LT. 0.OR.ISW(10) .EQ. -2) GO TO 1400 PLT17900
 1710 CONTINUE                                                          PLT18000
      IN = IABS(NETRAD)                                                 PLT18100
      IF (ISW(3) .EQ. 0) GO TO 1820                                     PLT18200
C     PLOT REL VAR OF MOMENTUM AND DEPTH VS. TIME                       PLT18300
      E2 = 1.0E5                                                        PLT18400
      DO 1810 J=1,2                                                     PLT18500
      E1 = 0.0                                                          PLT18600
      DO 1805 I=1,IN                                                    PLT18700
 1805 E1 = AMAX1(E1,VAR(I,J))                                           PLT18800
      DO 1806 I=1,IN                                                    PLT18900
      VAR(I,J) = VAR(I,J)/E1                                            PLT19000
```

```
      1806 E2 = AMIN1(E2,VAR(I,J))                                           PLT19100
      1810 CONTINUE                                                          PLT19200
           E1 = 1.0                                                          PLT19300
           CALL HEC004(0.0,PLTM(IN),E2,E1,4)                                 PLT19400
           CALL HEC007(PLTM,VAR,IN,1,E2)                                     PLT19500
           CALL HEC015(0,XM)                                                 PLT19600
      1820 CONTINUE                                                          PLT19700
           IF (ISW(5) .EQ. 0) GO TO 1900                                     PLT19800
           E2 = 1.0E5                                                        PLT19900
           DO 1850 J=1,2                                                     PLT20000
           E1 = 0.0                                                          PLT20100
           DO 1830 I=1,IN                                                    PLT20200
      1830 E1 = AMAX1(E1,VOL(I,J))                                           PLT20300
           DO 1840 I=1,IN                                                    PLT20400
           VOL(I,J) = VOL(I,J)/E1                                            PLT20500
      1840 E2 = AMIN1(E2,VOL(I,J))                                           PLT20600
      1850 CONTINUE                                                          PLT20700
           E1 = 1.0                                                          PLT20800
           CALL HEC004(0.0,PLTM(IN),E2,E1,5)                                 PLT20900
           CALL HEC007(PLTM,VOL,IN,2,E2)                                     PLT21000
           CALL HEC015(0,XM)                                                 PLT21100
      1900 CONTINUE                                                          PLT21200
      2000 FORMAT ('1=',E11.6,', 2=',E11.6,', 3=',E11.6)                     PLT21300
      2001 FORMAT ('WSMR - JUL/M/D/H=',I3,'/',I2,'/',I2,'/',I4,'Z, STEP ',I3,PLT21400
          1', TIME ',F7.2,' MIN')                                            PLT21500
      2002 FORMAT ('1',39X,'*-* SOLUTION FOR WSMR - JUL/M/D/H=',I3,'/',I2,'/',PLT21600
          1,I2,'/',I4,'Z *-*'/25X,'** WIND FIELD, LAYER HEIGHT, AND ',A6,A5,'PLT21700
          2 AT TIME STEP ',I4,' (',F7.2,' HOURS) **'/)                       PLT21800
      2003 FORMAT (5X,'X INDEX  Y INDEX   X COORDINATE  Y COORDINATE   U COMPPLT21900
          1ONENT   V COMPONENT    LAYER HEIGHT    ',A6,A5/8X,'I',8X,'J',8X,'(PLT22000
          2METERS)       (METERS)      (METERS/SEC)  (METERS/SEC)     (METERS)',PLT22100
          37X,A6,A4/1X,64('--'))                                             PLT22200
      2004 FORMAT (5X,'X INDEX  Y INDEX   X COORDINATE  Y COORDINATE   WIND SPLT22300
          1PEED       DIRECTION     LAYER HEIGHT    ',A6,A5/8X,'I',8X,'J',8X,'(PLT22400
          2METERS)       (METERS)      (METERS/SEC)    (DEGREES)      (METERS)',PLT22500
          37X,A6,A4/1X,64('--'))                                             PLT22600
      2005 FORMAT (1X,2I9,3X,2F14.3,2F14.6,2F16.6)                           PLT22700
           IF (ISW(2) .EQ. 0) GO TO 2090                                     PLT22800
```

```
      MFLGS = 0
      CALL BLKIN(INTO,H,1,WSTRRN,ISTS)
      CALL BLKIN(4090,NETRAD,1,TRDFFL,ISTS)
      IHOURS = IHOUR*100+NINUTE
      IF (KINDSO .EQ. 0) GO TO 9001
      I = 3
      IF (KINDSO .EQ. 1) I = 1
      IF (KINDSO .EQ. 2) I = 2
      ENCODE (60,9000,RUNID,J) IULIAN,IMONTH,IDAY,IHOURS,(IMTM(K),K=1,4)
     1,IMTN(I)
 9000 FORMAT ('*SMR- JUL/M/D/H=',I3,'/',I2,'/',I2,'/',I4,'Z, ',3A6,A3,A6
     1)
      GO TO 9002
 9001 ENCODE (60,9000,RUNID,J) IULIAN,IMONTH,IDAY,IHOURS,(IMTL(K),K=1,4)
     1,IMTN(3)
 9002 CONTINUE
      CALL HEC004(X(1),X(NX),Y(1),Y(NY),2)
      CALL HEC005(X(1),Y(1),M,H,P,0,N)
      XD = (X(NX)-X(1))*SCL*0.5
      CALL BLKIN(4090,NETRAD,1,TRDFFL,ISTS)
      IF (CNISS(1) .LE. 0.0) GO TO 2010
      DO 2006 L=1,3
 2006 DCONC(L,1) = CDONC(L,1)
      L = 0
 2007 L = L+1
      IF (L .GT. 3.OR.CNISS(L) .LE. 0.0) GO TO 2010
      DO 2009 I=1,NETRAD
      SIGY = 2.0*SIGMY(I)*SIGMY(I)
      A = CHIT(I)/CNISS(L)
      IF (A .LE. 1.0) GO TO 2008
      YCHIT(I,L) = CISO(SIGY,A)
      GO TO 2009
 2008 YCHIT(I,L) = 0.0
 2009 CONTINUE
      GO TO 2007
 2010 IF (KINDSO .NE. 0.OR.DSISS(1) .LE. 0.0) GO TO 2015
      DO 2011 L=1,3
 2011 DCONC(L,2) = CDONC(L,2)
```

```
      L = 0                                                          PLT26700
 2012 L = L+1                                                        PLT26800
      IF (L .GT. 3.OR.DSISS(L) .LE. 0.0) GO TO 2015                  PLT26900
      DO 2014 I=1,NETRAD                                             PLT27000
      SIGY = 2.0*SIGMY(I)*SIGMY(I)                                   PLT27100
      A = DOST(I)/DSISS(L)                                           PLT27200
      IF (A .LE. 1.0) GO TO 2013                                     PLT27300
      YDOST(I,L) = CISO(SIGY,A)                                      PLT27400
      GO TO 2014                                                     PLT27500
 2013 YDOST(I,L) = 0.0                                               PLT27600
 2014 CONTINUE                                                       PLT27700
      GO TO 2012                                                     PLT27800
 2015 CONTINUE                                                       PLT27900
      K1 = 2                                                         PLT28000
      IF (KINDSO .NE. 0) K1 = 1                                      PLT28100
      E3 = 0.05/SCL                                                  PLT28200
      DO 2080 K=1,K1                                                 PLT28300
      MFLGS = MFLGS+1                                                PLT28400
      CALL HEC016(MFLGS)                                             PLT28500
      IF (NETRAD .LE. 0) GO TO 2080                                  PLT28600
      Y1 = -0.9                                                      PLT28700
      X1 = 0.5*(XD+1.5)+XD-FLOAT(K-1)*4.1                            PLT28800
      CALL HEC014(X1,Y1,.075,IDLBL(1,K),0.0,23)                      PLT28900
C     LIST ISOPLETH LEVELS                                           PLT29000
      ENCODE (43,2000,ZELEVR,J) (DCONC(L,K),L=1,3)                   PLT29100
      X1 = 0.5*(XD+.275)+XD-FLOAT(K-1)*4.125                         PLT29200
      CALL HEC014(X1,Y1-.1,.075,ZELEVR,0.0,43)                       PLT29300
      DO 2017 L=1,3                                                  PLT29400
      DO 2016 I=1,NETRAD                                             PLT29500
      IF (COND(I,L,K) .GT. 0.0) GO TO 2016                           PLT29600
      IF (I-1 .GT. 0.AND.COND(I-1,L,K) .GT. 0.0) GO TO 2016          PLT29700
      IF (I+1 .LE. NETRAD.AND.COND(I+1,L,K) .GT. 0.0) GO TO 2016     PLT29800
      COND(I,L,K) = -1.0                                             PLT29900
 2016 CONTINUE                                                       PLT30000
 2017 CONTINUE                                                       PLT30100
      T = -1.0                                                       PLT30200
      DO 2070 J=1,2                                                  PLT30300
      T = T*-1.0                                                     PLT30400
```

```
      DO 2060 L=1,3                                                     PLT30500
      IF (DCONC(L,K) .LE. 0.0) GO TO 2070                               PLT30600
      XLST = XSOURC                                                     PLT30700
      YLST = YSOURC                                                     PLT30800
      LS = 3                                                            PLT30900
      I1 = 0                                                            PLT31000
 2018 I1 = I1+1                                                         PLT31100
      IF (COND(I1,L,K) .LE. 0.0) GO TO 2018                             PLT31200
      I1 = I1-1                                                         PLT31300
      IF (I1 .LT. 1) I1 = 1                                             PLT31400
      I2 = NETRAD+1                                                     PLT31500
 2019 I2 = I2-1                                                         PLT31600
      IF (COND(I2,L,K) .LE. 0.0) GO TO 2019                             PLT31700
      I2 = I2+1                                                         PLT31800
      IF (I2 .GT. NETRAD) I2 = NETRAD                                   PLT31900
      IF (I2 .LE. I1) GO TO 2060                                        PLT32000
      DO 2050 I=I1,I2                                                   PLT32100
      A1 = XX(I)-XLST                                                   PLT32200
      A2 = YY(I)-YLST                                                   PLT32300
      IF (A1) 2029,2028,2029                                            PLT32400
 2028 IF (A2) 2029,2030,2029                                            PLT32500
 2029 A2 = ATAN2(A2,A1)                                                 PLT32600
 2030 A1 = A2-1.5707963                                                 PLT32700
      E1 = COS(A1)*T                                                    PLT32800
      E2 = SIN(A1)*T                                                    PLT32900
      XP = COND(I,L,K)                                                  PLT33000
      YP = COND(I,L,K)                                                  PLT33100
      IF (COND(I,L,K) .GE. 0.0) GO TO 2032                              PLT33200
      XP = 0.0                                                          PLT33300
      YP = 0.0                                                          PLT33400
 2032 XP = (XX(I)-X(1)+XP*E1)*SCL                                       PLT33500
      YP = (YY(1)-Y(1)+YP*E2)*SCL                                       PLT33600
      XLST = XX(I)                                                      PLT33700
      YLST = YY(I)                                                      PLT33800
      IF (COND(I,L,K) .LT. 0.0) GO TO 2040                              PLT33900
      IF (XP .LT. 0.0.OR.XP .GT. XNCH) GO TO 2040                       PLT34000
      IF (YP .LT. 0.0.OR.YP .GT. YNCH) GO TO 2040                       PLT34100
      CALL PLOT(XP,YP,LS)                                               PLT34200
```

```
      LS = 2                                                          PLT34300
      IF (MOD(I,40) .NE. 0) GO TO 2050                                PLT34400
      E1 = (XX(I)-X(1)+(COND(I,L,K)+E3)*E1)*SCL                       PLT34500
      E2 = (YY(I)-Y(1)+(COND(I,L,K)+E3)*E2)*SCL                       PLT34600
      A1 = L                                                          PLT34700
      CALL NUMBER(E1,E2,.05,A1,0.0,-1)                                PLT34800
      CALL PLOT(XP,YP,3)                                              PLT34900
      GO TO 2050                                                      PLT35000
 2040 LS = 3                                                          PLT35100
 2050 CONTINUE                                                        PLT35200
 2060 CONTINUE                                                        PLT35300
 2070 CONTINUE                                                        PLT35400
 2080 CONTINUE                                                        PLT35500
      CALL HEC015(0,XM)                                               PLT35600
 2090 IF (IPLOT .NE. 0) CALL HEC015(1,XMIN)                           PLT35700
 3000 CONTINUE                                                        PLT35800
      IF (IPLOT .NE. 0) WRITE (XOTFIL,3001)                           PLT35900
 3001 FORMAT ('0 PLOT SECTION COMPLETED')                             PLT36000
      RETURN                                                          PLT36100
      END                                                             PLT36200
```

```
      SUBROUTINE BLKIN (NWDBLK, ISTART, NBKIN, LSFILE, ISTAT)           BIN00100
C     DISK VERSION. BLOCK TRANSFER FROM RANDOM ACCESS DISK FILE TO CORE.BIN00200
C BLKIN TRANSFERS TO CORE A BLOCK FROM A RANDOM ACCESS FILE THAT        BIN00300
C CONTAINS BLOCKS THAT ARE ALL OF THE SAME SIZE.                        BIN00400
C NWDBLK = NO. OF WORDS PER BLOCK IN THE FILE AND THE NO. OF WORDS TO BEBIN00500
C          TRANSFERRED TO CORE ON THIS CALL.                            BIN00600
C ISTART = STARTING ADDRESS IN CORE WHERE THE BLOCK IS TO BE TRANSFERREDBIN00700
C          TO.                                                          BIN00800
C NBKIN  = NO. OF THIS BLOCK IN THE FILE.  NBKIN = 1 IS THE FIRST BLOCK BIN00900
C          NO. IN THE FILE.                                             BIN01000
C LSFILE = LOGICAL SYSTEM FILE NO. (0-15).                              BIN01100
C ISTAT  = STATUS RETURNED TO USER.  ISTAT = 0 INDICATES NO ERRORS.     BIN01200
C          ISTAT = 1 INDICATES AN ERROR OF SOME KIND.                   BIN01300
C                                                                       BIN01400
C 1108 DISK VERSION                                                     BIN01500
C                                                                       BIN01600
C RESTRICTIONS ON THIS VERSION OF BLKIN                                 BIN01700
C                                                                       BIN01800
C THE STATUS ISTAT RETURNED TO THE USER WILL ALWAYS BE ZERO SINCE THE   BIN01900
C FSTRD ROUTINE DOES NOT RETURN ANY STATUS INFORMATION.  FSTRD HAS IT'S BIN02000
C OWN ERROR MESSAGES.                                                   BIN02100
C                                                                       BIN02200
      NSECPB=(NWDBLK+27)/28                                             BIN02300
      NBKM1=NBKIN-1                                                     BIN02400
      CALL FSTRD (NWDBLK, ISTART, NSECPB, NBKM1, 0, LSFILE)             BIN02500
      ISTAT=0                                                           BIN02600
      RETURN                                                            BIN02700
      END                                                               BIN02800
```

```
      SUBROUTINE BLKOUT (NWDBLK, ISTART, NBKOUT, LSFILE, ISTAT)         BOT00100
C     DISK VERSION. BLOCK TRANSFER FROM CORE TO RANDOM ACCESS DISK FILE.BOT00200
C BLKOUT TRANSFERS A BLOCK FROM CORE TO A RANDOM ACCESS FILE WHICH      BOT00300
C CONTAINS BLOCKS THAT ARE ALL OF THE SAME SIZE.                        BOT00400
C NWDBLK = NO. OF WORDS PER BLOCK IN THE FILE AND THE NO. OF WORDS TO BEBOT00500
C          TRANSFERRED FROM CORE ON THIS CALL.                          BOT00600
C ISTART = STARTING ADDRESS IN CORE WHERE THE BLOCK IS TO BE TRANSFERREDBOT00700
C          FROM.                                                        BOT00800
C NBKOUT = NO. OF THIS BLOCK IN THE FILE.  NBKOUT = 1 IS THE FIRST BLOCKBOT00900
C          NO. IN THE FILE.                                             BOT01000
C LSFILE = LOGICAL SYSTEM FILE NO. (0-15).                              BOT01100
C ISTAT  = STATUS RETURNED TO USER.  ISTAT = 0 INDICATES NO ERRORS.     BOT01200
C          ISTAT = 1 INDICATES AN ERROR OF SOME KIND.                   BOT01300
C                                                                       BOT01400
C 1108 DISK VERSION                                                     BOT01500
C                                                                       BOT01600
C RESTRICTIONS ON THIS VERSION OF BLKOUT                                BOT01700
C                                                                       BOT01800
C THE STATUS ISTAT RETURNED TO THE USER WILL ALWAYS BE ZERO SINCE THE   BOT01900
C FSTWT ROUTINE DOES NOT RETURN ANY STATUS INFORMATION.  FSTWT HAS IT'S BOT02000
C OWN ERROR MESSAGES.                                                   BOT02100
C                                                                       BOT02200
      NSECPB=(NWDBLK+27)/28                                             BOT02300
      NBKM1=NBKOUT-1                                                    BOT02400
      CALL FSTWT (NWDBLK, ISTART, NSECPB, NBKM1, 0, LSFILE)             BOT02500
      ISTAT=0                                                           BOT02600
      RETURN                                                            BOT02700
      END                                                               BOT02800
```

```
      SUBROUTINE HEC003(X,Y)                                            H0300100
C     THIS SUBROUTINE CONVERTS THE U AND V COMPONENTS (X AND Y) OF THE  H0300200
C     WIND SPEED INTO WIND SPEED AND DIRECTION (X AND Y)                H0300300
      DATA RAD/57.29582/                                                H0300400
      IF (X) 20,10,20                                                   H0300500
   10 IF (Y) 20,30,20                                                   H0300600
   20 CONTINUE                                                          H0300700
      DIR = 270.0-ATAN2(Y,X)*RAD                                        H0300800
      IF (DIR .GE. 360.0) DIR = DIR-360.0                               H0300900
      X = SQRT(X*X+Y*Y)                                                 H0301000
      Y = DIR                                                           H0301100
   30 CONTINUE                                                          H0301200
      RETURN                                                            H0301300
      END                                                               H0301400
```

```
      SUBROUTINE HEC004(XMN,XMX,YMN,YMX,JSW)                            H0400100
      INCLUDE N1,LIST                                                   H0400200
      COMMON /DTA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0400300
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     H0400400
      COMMON/BS/LBL1(5),LBL2(7),LBL3(7),LBL6(5),LBL9(2),LBL4(4),LBL5(3) H0400500
      EQUIVALENCE (J3,DX),(J4,DY),(J5,J),(J6,NCH),(J7,YPL),(J8,XPL),    H0400600
     1(J9,TIC),(J10,IB),(J11,X),(J12,Y)                                 H0400700
      INTEGER BLNK,RUNID                                                H0400800
      DATA BLNK/' '/                                                    H0400900
      DATA LBL1/'DISTANCE (M)   (SOUTH-NORTH)'/                         H0401000
      DATA LBL2/'NORMALIZED REL VAR OF CHANGE IN MOMENTUM'/             H0401100
      DATA LBL3/'NORMALIZED REL VAR OF CHANGE IN DEPTH'/                H0401200
      DATA LBL4/'NORMALIZED MOMENTUM'/                                  H0401300
      DATA LBL5/'NORMALIZED VOLUME'/                                    H0401400
      DATA LBL6/'DISTANCE (M)   (WEST-EAST)'/                           H0401500
      DATA LBL9/'TIME (SEC)'/                                           H0401600
C     THIS SUBROUTINE DRAWS AND LABELS THE PLOT AXES                    H0401700
      IF (JSW .GE. 4) GO TO 10                                          H0401800
      DX = 1000.                                                        H0401900
      DY = 1000.                                                        H0402000
      J1 = -1                                                           H0402100
      J2 = -1                                                           H0402200
      XNCH = SCL*(XMX-XMN)                                              H0402300
      YNCH = SCL*(YMX-YMN)                                              H0402400
      SCLX = SCL                                                        H0402500
      SCLY = SCL                                                        H0402600
      GO TO 20                                                          H0402700
   10 DX = (XMX-XMN)/50.0                                               H0402800
      DY = (YMX-YMN)/50.0                                               H0402900
      J1 = 2                                                            H0403000
      XNCH = 10.0                                                       H0403100
      YNCH = 7.0                                                        H0403200
      SCLX = XNCH/(XMX-XMN)                                             H0403300
      SCLY = YNCH/(YMX-YMN)                                             H0403400
      J2 = 6                                                            H0403500
   20 CONTINUE                                                          H0403600
      XM = XNCH+2.0                                                     H0403700
      YM = YNCH+2.0                                                     H0403800
```

```
      IF (XM .GT. 33.0) XM = 33.0                                       H0403900
      IF (YM .GT. 33.0) YM = 33.0                                       H0404000
      IF (XNCH .GT. 31.0) XNCH = 31.0                                   H0404100
      IF (YNCH .GT. 31.0) YNCH = 31.0                                   H0404200
      CALL HEC013(XM,YM,ISW(20))                                        H0404300
      CALL PLOT(0.0,0.0,3)                                              H0404400
      CALL PLOT(0.0,YNCH,2)                                             H0404500
      CALL PLOT(XNCH,YNCH,2)                                            H0404600
      CALL PLOT(XNCH,0.0,2)                                             H0404700
      CALL PLOT(0.0,0.0,2)                                              H0404800
      YPL = 0.0                                                         H0404900
      TIC = 1.0                                                         H0405000
      DO 70 J=1,2                                                       H0405100
      IB = 9                                                            H0405200
      X = XMN-DX                                                        H0405300
   40 X = X+DX                                                          H0405400
      IB = IB+1                                                         H0405500
      XPL = (X-XMN)*SCLX                                                H0405600
      IF (XPL .LT. XNCH) GO TO 50                                       H0405700
      XPL = XNCH                                                        H0405800
   50 CALL PLOT(XPL,YPL,3)                                              H0405900
      IF (IB .LT. 10) GO TO 60                                          H0406000
      IB = 0                                                            H0406100
      CALL PLOT(XPL,YPL+.1*TIC,2)                                       H0406200
      IF (J .EQ. 2) GO TO 61                                            H0406300
      CALL NUMBER(XPL-.225,YPL-.1,.075,X,0.0,J1)                        H0406400
      GO TO 61                                                          H0406500
   60 CALL PLOT(XPL,YPL+.05*TIC,2)                                      H0406600
   61 IF (XPL .LT. XNCH) GO TO 40                                       H0406700
      YPL = YNCH                                                        H0406800
      TIC = -1.0                                                        H0406900
   70 CONTINUE                                                          H0407000
      XPL = 0.0                                                         H0407100
      TIC = 1.0                                                         H0407200
      DO 110 J=1,2                                                      H0407300
      IB = 9                                                            H0407400
      Y = YMN-DY                                                        H0407500
   80 Y = Y+DY                                                          H0407600
```

```
      IB = IB+1                                                         H0407700
      YPL = (Y-YMN)*SCLY                                                H0407800
      IF (YPL .LT. YNCH) GO TO 90                                       H0407900
      YPL = YNCH                                                        H0408000
   90 CALL PLOT(XPL,YPL,3)                                              H0408100
      IF (IB .LT. 10) GO TO 100                                         H0408200
      IB = 0                                                            H0408300
      CALL PLOT(XPL+.1*TIC,YPL,2)                                       H0408400
      IF (J .EQ. 2) GO TO 101                                           H0408500
      CALL NUMBER(XPL-.1,YPL-.23,.075,Y,90.0,J2)                        H0408600
      GO TO 101                                                         H0408700
  100 CALL PLOT(XPL+.05*TIC,YPL,2)                                      H0408800
  101 IF (YPL .LT. YNCH) GO TO 80                                       H0408900
      XPL = XNCH                                                        H0409000
      TIC = -1.0                                                        H0409100
  110 CONTINUE                                                          H0409200
C     LABEL AXES                                                        H0409300
      IF (JSW .GE. 4) GO TO 130                                         H0409400
      X = -.3                                                           H0409500
      Y = 0.5*(YNCH-2.7)                                                H0409600
      CALL HEC014(X,Y,.1,LBL1,90.0,27)                                  H0409700
      X = 0.5*(XNCH-2.5)                                                H0409800
      Y = -.3                                                           H0409900
      CALL HEC014(X,Y,.1,LBL6,0.0,25)                                   H0410000
      GO TO 140                                                         H0410100
  130 X = 0.5*(XNCH-1.0)                                                H0410200
      Y = -.3                                                           H0410300
      CALL HEC014(X,Y,.1,LBL9,0.0,10)                                   H0410400
  140 NCH = 66                                                          H0410500
      J = 11                                                            H0410600
  150 NCH = NCH-6                                                       H0410700
      J = J-1                                                           H0410800
      IF (J .EQ. 0) GO TO 160                                           H0410900
      IF (RUNID(J) .EQ. BLNK) GO TO 150                                 H0411000
      X = 0.5*(XNCH-0.1*FLOAT(NCH))                                     H0411100
      Y = -.7                                                           H0411200
      CALL HEC014(X,Y,.1,RUNID,0.0,NCH)                                 H0411300
  160 CONTINUE                                                          H0411400
```

```
      XM = XNCH+2.0                                              H0411500
      YM = YNCH+2.0                                              H0411600
      MFLGS = 0                                                  H0411700
      RETURN                                                     H0411800
      END                                                        H0411900
```

```
      COMPILER (XM=1)                                                   H0500100
      SUBROUTINE HEC005(X,Y,Z,H,P,JSW,W)                                H0500200
      INCLUDE N1,LIST                                                   H0500300
      COMMON /DTA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0500400
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     H0500500
      DIMENSION X(1),Y(1),Z(1),H(IDIM,1),P(IDIM,1)                      H0500600
      DIMENSION W(1)                                                    H0500700
C     THIS SUBROUTINE SETS UP ARRAYS FOR CONTOUR PLOTTING               H0500800
      J6 = IND                                                          H0500900
      J8 = J6-IST+1                                                     H0501000
   10 IF ((X(J8)-X(1))*SCL .LE. XNCH) GO TO 20                          H0501100
      J8 = J8-1                                                         H0501200
      J6 = J6-1                                                         H0501300
      IF (J6 .LE. IST) GO TO 80                                         H0501400
      GO TO 10                                                          H0501500
   20 J7 = JND                                                          H0501600
      J9 = J7-JST+1                                                     H0501700
   30 IF ((Y(J9)-Y(1))*SCL .LE. YNCH) GO TO 40                          H0501800
      J9 = J9-1                                                         H0501900
      J7 = J7-1                                                         H0502000
      IF (J7 .LE. JST) GO TO 80                                         H0502100
      GO TO 30                                                          H0502200
   40 IF (JSW .NE. 0) GO TO 60                                          H0502300
      J2 = 0                                                            H0502400
      DO 50 J=JST,J7                                                    H0502500
      J2 = J2+1                                                         H0502600
      J1 = (J2-1)*J8                                                    H0502700
      DO 50 I=IST,J6                                                    H0502800
      J1 = J1+1                                                         H0502900
      Z(J1) = H(I,J)                                                    H0503000
   50 CONTINUE                                                          H0503100
      CALL HEC009(X,Y,Z,J8,J9,CONT,NCONT,SCL,W,0,W(J8*J9+1),Z(J8*J9+1)) H0503200
      GO TO 80                                                          H0503300
   60 CONTINUE                                                          H0503400
      J2 = 0                                                            H0503500
      DO 70 J=JST,J7                                                    H0503600
      J2 = J2+1                                                         H0503700
      J1 = (J2-1)*J8                                                    H0503800
```

```
      DO 70 I=IST,J6                                                    H0503900
      J1 = J1+1                                                         H0504000
      A = 0.0                                                           H0504100
      IF (ISW(7) .NE. 0) A = H(I,J)                                     H0504200
      Z(J1) = P(I,J)+A                                                  H0504300
   70 CONTINUE                                                          H0504400
      IF (MFLGS .NE. 0) CALL HEC016(MFLGS)                              H0504500
      CALL HEC009(X,Y,Z,J8,J9,CONTS,NCONTS,SCL,W,ISW(10),W(J8*J9+1),Z(J8H0504600
     1*J9+1))                                                           H0504700
   80 CONTINUE                                                          H0504800
      RETURN                                                            H0504900
      END                                                               H0505000
```

```
      SUBROUTINE HEC006(X,Y,U,V,NX,NY,P,XMIN)                          H0600100
      INCLUDE N1,LIST                                                  H0600200
C     THIS SUBROUTINE DRAWS VECTORS AT EACH GRID COORDINATE            H0600300
      COMMON /DIA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0600400
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE    H0600500
      DIMENSION X(1),Y(1),U(NX,1),V(NX,1),XV(3),YV(3)                  H0600600
      EQUIVALENCE (J1,RSCL),(J2,PSCL),(J3,YP),(J4,XP),(J5,X1),(J6,Y1), H0600700
     1(J7,XV),(J10,YV),(J13,X2),(J14,Y2)                               H0600800
      IF (MFLGS .NE. 0) CALL HEC016(MFLGS)                             H0600900
C     CALC MAX VELOCITY                                                H0601000
      UVMAX = 0.0                                                      H0601100
      DO 20 J=1,NY                                                     H0601200
      DO 20 I=1,NX                                                     H0601300
   20 UVMAX = AMAX1(UVMAX,U(I,J)*U(I,J)+V(I,J)*V(I,J))                 H0601400
      UVMAX = SQRT(UVMAX)                                              H0601500
C     CALC VECTOR SCALE FACTOR IN INCHES PER METER/SEC                 H0601600
      RSCL = .80*SCL*XMIN/UVMAX                                        H0601700
C     PLOT VECTOR SCALE 1 INCH = PSCL M/S                              H0601800
      PSCL = 1.0/RSCL                                                  H0601900
C     CALC AND DRAW VECTORS                                            H0602000
      DO 60 J=1,NY                                                     H0602100
      YP = (Y(J)-Y(1))*SCL                                             H0602200
      DO 60 I=1,NX                                                     H0602300
      XP = (X(I)-X(1))*SCL                                             H0602400
      X1 = U(I,J)*RSCL                                                 H0602500
      Y1 = V(I,J)*RSCL                                                 H0602600
      X2 = X1                                                          H0602700
      Y2 = Y1                                                          H0602800
      CALL HEC003(X2,Y2)                                               H0602900
      IF (X2 .LE. 0.0) GO TO 60                                        H0603000
      Y2 = (Y2+180.)*RAD                                               H0603100
      X2 = COS(Y2)                                                     H0603200
      Y2 = SIN(Y2)                                                     H0603300
      YV(1) = SQRT(X1*X1+Y1*Y1)                                        H0603400
      XV(1) = 0.                                                       H0603500
      YV(2) = YV(1)-.05                                                H0603600
      XV(2) = .025                                                     H0603700
      YV(3) = YV(2)                                                    H0603800
```

```
      XV(3) = -XV(2)                                                    H0603900
      DO 50 K=1,3                                                       H0604000
      X1 = YV(K)*Y2-XV(K)*X2                                            H0604100
      Y1 = XV(K)*Y2+YV(K)*X2                                            H0604200
      XV(K) = X1                                                        H0604300
   50 YV(K) = Y1                                                        H0604400
      CALL PLOT(XP,YP,3)                                                H0604500
      CALL PLOT(XP+XV(1),YP+YV(1),2)                                    H0604600
      CALL PLOT(XP+XV(2),YP+YV(2),3)                                    H0604700
      CALL PLOT(XP+XV(1),YP+YV(1),2)                                    H0604800
      CALL PLOT(XP+XV(3),YP+YV(3),2)                                    H0604900
   60 CONTINUE                                                          H0605000
      CALL HEC014(0.0,-0.5,.075,22H(VECTOR SCALE 1 INCH =,0.0,22)       H0605100
      CALL NUMBER(1.65,-0.5,.075,PSCL,0.0,4)                            H0605200
      CALL HEC014(2.175,-0.5,.075,5H M/S),0.0,5)                        H0605300
      CALL HEC014(0.0,-0.7,.075,15H(MAX VELOCITY =,0.0,15)              H0605400
      CALL NUMBER(1.2,-0.7,.075,UVMAX,0.0,3)                            H0605500
      CALL HEC014(1.8,-0.7,.075,6HM/SEC),0.0,6)                         H0605600
      IF (NCRV .EQ. 0) GO TO 70                                         H0605700
      CALL HEC008(U,V,X,Y,NX,NY,UVMAX,XMIN,NCRV,XST,YST,IEVEN,P)        H0605800
   70 CONTINUE                                                          H0605900
      RETURN                                                            H0606000
      END                                                               H0606100
```

```
      SUBROUTINE HEC007(TIME,VARVOL,L,K,YMIN)                           H0700100
      DIMENSION TIME(1),VARVOL(200,1)                                   H0700200
      COMMON/BS/LBL1(5),LBL2(7),LBL3(7),LBL6(5),LBL9(2),LBL4(4),LBL5(3) H0700300
      COMMON /DTA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0700400
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     H0700500
      EQUIVALENCE (J4,X),(J5,Y)                                         H0700600
C     THIS SUBROUTINE PLOTS THE RELATIVE VARIANCE OF THE CHANGE IN      H0700700
C     MOMENTUM VERSUS TIME, THE RELATIVE VARIANCE OF THE CHANGE IN DEPTHH0700800
C     VS. TIME, THE NORMALIZED MOMENTUM VS. TIME AND THE NORMALIZED     H0700900
C     VOLUME VS. TIME.                                                  H0701000
      DO 50 J=1,2                                                       H0701100
      CALL PLOT(0.0,0.0,3)                                              H0701200
      IF (J .EQ. 2) CALL HEC016(1)                                      H0701300
      DO 10 I=1,L                                                       H0701400
      X = TIME(I)*SCLX                                                  H0701500
      Y = (VARVOL(I,J)-YMIN)*SCLY                                       H0701600
      CALL PLOT(X,Y,2)                                                  H0701700
      IF (J .EQ. 2) GO TO 10                                            H0701800
      CALL PLOT(X,0.2,3)                                                H0701900
      CALL PLOT(X,0.4,2)                                                H0702000
      CALL PLOT(X,Y,3)                                                  H0702100
   10 CONTINUE                                                          H0702200
      IF (J .EQ. 2) GO TO 30                                            H0702300
      X = -0.3                                                          H0702400
      IF (K .EQ. 2) GO TO 20                                            H0702500
      Y = 0.5*(YNCH-4.0)                                                H0702600
      CALL HEC014(X,Y,.1,LBL2,90.0,40)                                  H0702700
      GO TO 50                                                          H0702800
   20 Y = 0.5*(YNCH-1.9)                                                H0702900
      CALL HEC014(X,Y,.1,LBL4,90.0,19)                                  H0703000
      GO TO 50                                                          H0703100
   30 X = -0.5                                                          H0703200
      IF (K .EQ. 2) GO TO 40                                            H0703300
      Y = 0.5*(YNCH-3.7)                                                H0703400
      CALL HEC014(X,Y,.1,LBL3,90.0,37)                                  H0703500
      GO TO 50                                                          H0703600
   40 Y = 0.5*(YNCH-1.7)                                                H0703700
      CALL HEC014(X,Y,.1,LBL5,90.0,17)                                  H0703800
```

```
      50 CONTINUE                                                       H0703900
         RETURN                                                         H0704000
         END                                                            H0704100
```

```
      SUBROUTINE HEC008(U,V,X,Y,NX,NY,UVMX,XMIN,MCRV,XST,YST,IEVEN,P)   H0800100
      INCLUDE FILES,LIST                                                H0800200
      DIMENSION U(NX,1),V(NX,1),X(1),Y(1),XST(1),YST(1),P(NX,1)         H0800300
      COMMON /DTA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0800400
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     H0800500
      COMMON /M1/ S(124)                                                H0800600
      EQUIVALENCE (S(1),LINE),(S(2),DX),(S(3),T),(S(4),N2),(S(5),N),    H0800700
     1(S(6),K2),(S(7),XP),(S(8),YP),(S(9),HXYM),(S(10),HXM),            H0800800
     2(S(11),HX),(S(12),HY),(S(13),HX1),(S(14),HY1),(S(15),HX5),        H0800900
     3(S(16),HY5),(S(17),HX21),(S(18),HY21),(S(19),HXY4),(S(20),N1),    H0801000
     4(S(21),K1),(S(22),IB),(S(23),HM),(S(24),DT),(S(25),X0),           H0801100
     5(S(26),Y0),(S(27),PX),(S(28),K),(S(29),HXN),(S(30),HXN1),         H0801200
     6(S(31),HYN),(S(32),HYN1),(S(33),NT),(S(34),X1),(S(35),Y1),        H0801300
     7(S(36),PX1),(S(37),PY),(S(38),PY1),(S(39),UX1),(S(40),UX),        H0801400
     8(S(41),UXX),(S(42),UY1),(S(43),UY),(S(44),UXY1),(S(45),UXY3),     H0801500
     9(S(46),UXY),(S(47),VX1),(S(48),VX),(S(49),VXX),(S(50),VY1)        H0801600
      EQUIVALENCE (S(51),VY),(S(52),VYY),(S(53),VXY1),(S(54),VXY3),     H0801700
     1(S(55),VXY),(S(56),UYY),(S(57),AH),(S(58),AK),(S(59),UU),         H0801800
     2(S(60),VV),(S(61),UU1),(S(62),VV1),(S(63),HMI),(S(64),XD),        H0801900
     3(S(65),S1),(S(66),S2),(S(67),UUL),(S(68),VVL),(S(69),U1),         H0802000
     4(S(70),U2),(S(71),V1),(S(72),V2)                                  H0802100
C     THIS SUBROUTINE CALCULATES AND DRAWS ALL TRAJECTORIES.            H0802200
      NCRV = IABS(MCRV)                                                 H0802300
      N2 = NX-1                                                         H0802400
      K2 = NY-1                                                         H0802500
      DX = 0.1*XMIN                                                     H0802600
      UUL = UVMX                                                        H0802700
      VVL = UVMX                                                        H0802800
      XP = XNCH-3.68                                                    H0802900
      YP = -.5                                                          H0803000
      CALL HEC014(XP,YP,.075,40H(DISTANCE BETWEEN TRAJECTORY TIC MARKS =H0803100
     1,0.0,40)                                                          H0803200
      X1 = 5.0*DX                                                       H0803300
      CALL NUMBER(XP+3.038,YP,.075,X1,0.0,-1)                           H0803400
      CALL HEC014(XP+3.375,YP,.075,4H(M)),0.0,4)                        H0803500
      HXYM = XMIN                                                       H0803600
      HXM = 0.05*HXYM                                                   H0803700
      IF (IEVEN .EQ. 0) GO TO 40                                        H0803800
```

```
      HX = X(2)-X(1)                                                    H0803900
      HY = Y(2)-Y(1)                                                    H0804000
      HX1 = 0.5/HX                                                      H0804100
      HY1 = 0.5/HY                                                      H0804200
      HX5 = .5*HX                                                       H0804300
      HY5 = .5*HY                                                       H0804400
      HX21 = 2.0*HX1*HX1                                                H0804500
      HY21 = 2.0*HY1*HY1                                                H0804600
      HXY4 = HX1*HY1                                                    H0804700
   40 CONTINUE                                                          H0804800
      DO 200 L=1,NCRV                                                   H0804900
      U1 = 0.0                                                          H0805000
      V1 = 0.0                                                          H0805100
      U2 = 0.0                                                          H0805200
      V2 = 0.0                                                          H0805300
      XD = 0.0                                                          H0805400
      DT = DX/(0.5*UVMX)                                                H0805500
      T = 0.0                                                           H0805600
      N1 = 0                                                            H0805700
      K1 = 0                                                            H0805800
      IB = 0                                                            H0805900
      XO = XST(L)                                                       H0806000
      YO = YST(L)                                                       H0806100
      IF (XO .GT. X(NX).OR.XO .LT. X(1)) GO TO 200                      H0806200
      IF (YO .GT. Y(NY).OR.YO .LT. Y(1)) GO TO 200                      H0806300
      XP = (XO-X(1))*SCLX                                               H0806400
      YP = (YO-Y(1))*SCLY                                               H0806500
      X1 = L                                                            H0806600
      CALL NUMBER(XP+.05,YP,.075,X1,0.0,-1)                             H0806700
      CALL PLOT(XP,YP+.02,3)                                            H0806800
      CALL PLOT(XP,YP,2)                                                H0806900
      CALL PLOT(XP+.01176,YP-.01618,2)                                  H0807000
      CALL PLOT(XP+.01902,YP+.0061803,3)                                H0807100
      CALL PLOT(XP,YP,2)                                                H0807200
      CALL PLOT(XP-.01176,YP-.01618,2)                                  H0807300
      CALL PLOT(XP-.01902,YP+.0061803,3)                                H0807400
      CALL PLOT(XP,YP,2)                                                H0807500
   60 IF (IEVEN .EQ. 0) GO TO 70                                        H0807600
```

```
      N = INT((XO-X(1))*HX1*2.0)+1                                H0807700
      IF (XO-X(N) .GT. HX5) N = N+1                               H0807800
      IF (N .GE. NX) N = NX-1                                     H0807900
      IF (N .LE. 1) N = 2                                         H0808000
      K = INT((YO-Y(1))*HY1*2.0)+1                                H0808100
      IF (YO-Y(K) .GT. HY5) K = K+1                               H0808200
      IF (K .GE. NY) K=NY-1                                       H0808300
      IF (K .LE. 1) K = 2                                         H0808400
      GO TO 120                                                   H0808500
   70 IF (XO-1.0 .LT. X(1).OR.XO+1.0 .GT. X(NX)) GO TO 200        H0808600
      IF (YO-1.0 .LT. Y(1).OR.YO+1.0 .GT. Y(NY)) GO TO 200        H0808700
      N = 2                                                       H0808800
   80 IF (XO .LT. X(N)) GO TO 90                                  H0808900
      N = N+1                                                     H0809000
      GO TO 80                                                    H0809100
   90 IF (X(N)-XO .GT. XO-X(N-1)) N = N-1                         H0809200
      IF (N .LE. 1) N = 2                                         H0809300
      IF (N .GE. NX) N = NX-1                                     H0809400
      K = 2                                                       H0809500
  100 IF (YO .LT. Y(K)) GO TO 110                                 H0809600
      K = K+1                                                     H0809700
      GO TO 100                                                   H0809800
  110 IF (Y(K)-YO .GT. YO-Y(K-1)) K = K-1                         H0809900
      IF (K .LE. 1) K = 2                                         H0810000
      IF (K .GE. NY) K = NY-1                                     H0810100
      HXN = X(N)-X(N-1)                                           H0810200
      HXN1 = X(N+1)-X(N)                                          H0810300
      HYN = Y(K)-Y(K-1)                                           H0810400
      HYN1 = Y(K+1)-Y(K)                                          H0810500
  120 CONTINUE                                                    H0810600
C     CALC NEXT UU,VV                                             H0810700
      NT = 0                                                      H0810800
      X1 = XO                                                     H0810900
      Y1 = YO                                                     H0811000
      IF (N1 .EQ. N.AND.K1 .EQ. K) GO TO 140                      H0811100
      N1 = N                                                      H0811200
      K1 = K                                                      H0811300
      IF (IEVEN .NE. 0) GO TO 130                                 H0811400
```

```
      PX = 1.0/((HXN+HXN1)*HXN)                                           H0811500
      PX1 = 1.0/((HXN+HXN1)*HXN1)                                         H0811600
      PY = 1.0/((HYN+HYN1)*HYN)                                           H0811700
      PY1 = 1.0/((HYN+HYN1)*HYN1)                                         H0811800
      UX1 = PX1*(U(N+1,K)-U(N,K))                                         H0811900
      UX = PX*(U(N,K)-U(N-1,K))                                           H0812000
      UXX = 2.0*(UX1-UX)                                                  H0812100
      UX = HXN*UX1+HXN1*UX                                                H0812200
      UY1 = PY1*(U(N,K+1)-U(N,K))                                         H0812300
      UY = PY*(U(N,K)-U(N,K-1))                                           H0812400
      UYY = 2.0*(UY1-UY)                                                  H0812500
      UY = HYN*UY1+HYN1*UY                                                H0812600
      UXY1 = PX1*HXN*(U(N+1,K-1)-U(N,K-1))+PX*HXN1*(U(N,K-1)-U(N-1,K-1))H0812700
      UXY3 = PX1*HXN*(U(N+1,K+1)-U(N,K+1))+PX*HXN1*(U(N,K+1)-U(N-1,K+1))H0812800
      UXY = PY1*HYN*(UXY3-UX)+PY*HYN1*(UX-UXY1)                           H0812900
      VX1 = PX1*(V(N+1,K)-V(N,K))                                         H0813000
      VX = PX*(V(N,K)-V(N-1,K))                                           H0813100
      VXX = 2.0*(VX1-VX)                                                  H0813200
      VX = HXN*VX1+HXN1*VX                                                H0813300
      VY1 = PY1*(V(N,K+1)-V(N,K))                                         H0813400
      VY = PY*(V(N,K)-V(N,K-1))                                           H0813500
      VYY = 2.0*(VY1-VY)                                                  H0813600
      VY = HYN*VY1+HYN1*VY                                                H0813700
      VXY1 = PX1*HXN*(V(N+1,K-1)-V(N,K-1))+PX*HXN1*(V(N,K-1)-V(N-1,K-1))H0813800
      VXY3 = PX1*HXN*(V(N+1,K+1)-V(N,K+1))+PX*HXN1*(V(N,K+1)-V(N-1,K+1))H0813900
      VXY = PY1*HYN*(VXY3-VX)+PY*HYN1*(VX-VXY1)                           H0814000
      GO TO 140                                                           H0814100
  130 CONTINUE                                                            H0814200
      UX = (U(N+1,K)-U(N-1,K))*HX1                                        H0814300
      UY = (U(N,K+1)-U(N,K-1))*HY1                                        H0814400
      UXX = HX21*(U(N-1,K)-U(N,K)-U(N,K)+U(N+1,K))                        H0814500
      UYY = HY21*(U(N,K-1)-U(N,K)-U(N,K)+U(N,K+1))                        H0814600
      UXY = HXY4*(U(N+1,K+1)-U(N-1,K+1)-U(N+1,K-1)+U(N-1,K-1))            H0814700
      VX = (V(N+1,K)-V(N-1,K))*HX1                                        H0814800
      VY = (V(N,K+1)-V(N,K-1))*HY1                                        H0814900
      VXX = HX21*(V(N-1,K)-V(N,K)-V(N,K)+V(N+1,K))                        H0815000
      VYY = HY21*(V(N,K-1)-V(N,K)-V(N,K)+V(N,K+1))                        H0815100
      VXY = HXY4*(V(N+1,K+1)-V(N-1,K+1)-V(N+1,K-1)+V(N-1,K-1))            H0815200
```

```
  140 CONTINUE                                                           H0815300
      AH = X1-X(N)                                                       H0815400
      AK = Y1-Y(K)                                                       H0815500
      UU = U(N,K)+AH*UX+AK*UY+AH*AH*UXX+AH*AK*UXY+AK*AK*UYY              H0815600
      VV = V(N,K)+AH*VX+AK*VY+AH*AH*VXX+AH*AK*VXY+AK*AK*VYY              H0815700
      IF (NT .NE. 0) GO TO 150                                           H0815800
      NT = 3                                                             H0815900
      X1 = X1+UU*DT                                                      H0816000
      Y1 = Y1+VV*DT                                                      H0816100
      UU1 = UU                                                           H0816200
      VV1 = VV                                                           H0816300
      GO TO 140                                                          H0816400
  150 CONTINUE                                                           H0816500
      UU = 0.5*(UU+UU1)                                                  H0816600
      VV = 0.5*(VV+VV1)                                                  H0816700
  160 IF (IB .NE. 0) GO TO 170                                           H0816800
      GO TO 175                                                          H0816900
  165 HMI = HM                                                           H0817000
      X1 = SQRT(UU1*UU1+VV1*VV1)                                         H0817100
      LINE = 8                                                           H0817200
      IF (MCRV .GT. 0) GO TO 166                                         H0817300
      WRITE (XOTFIL,2000) L,DX                                           H0817400
      WRITE (XOTFIL,2001) IB,XO,YO,UU1,VV1,XD,X1,T,T,HMI                 H0817500
  166 CONTINUE                                                           H0817600
      IB = IB+1                                                          H0817700
  170 IF (UU) 173,171,173                                                H0817800
  171 IF (VV) 173,172,173                                                H0817900
  172 UU = UUL                                                           H0818000
      VV = VVL                                                           H0818100
  173 X1 = SQRT(UU*UU+VV*VV)                                             H0818200
      UUL = UU                                                           H0818300
      VVL = VV                                                           H0818400
      DT = DX/X1                                                         H0818500
      XO = XO+DT*UU                                                      H0818600
      YO = YO+DT*VV                                                      H0818700
      IF (XO.LT.X(1).OR.XO.GT.X(NX).OR.YO.LT.Y(1).OR.YO.GT.Y(NY)) GO TO H0818800
     1200                                                                H0818900
      T = T+DT                                                           H0819000
```

```
      XD = XD+DX
  175 IF (XO .GE. X(N2).AND.XO .LE. X(N2+1)) GO TO 177
      N2 = NX-1
  176 IF (XO .GE. X(N2)) GO TO 177
      N2 = N2-1
      IF (N2 .GT. 1) GO TO 176
  177 IF (YO .GE. Y(K2).AND.YO .LE. Y(K2+1)) GO TO 179
      K2 = NY-1
  178 IF (YO .GE. Y(K2)) GO TO 179
      K2 = K2-1
      IF (K2 .GT. 1) GO TO 178
  179 XP = (XO-X(N2))/(X(N2+1)-X(N2))
      YP = (YO-Y(K2))/(Y(K2+1)-Y(K2))
      HM = (1.0-XP)*(1.0-YP)*P(N2,K2)+XP*(1.0-YP)*P(N2+1,K2)+
     1(1.0-XP)*YP*P(N2,K2+1)+XP*YP*P(N2+1,K2+1)
      IF (IB .EQ. 0) GO TO 165
      HMI = 0.5*(HM+HMI)
      IF (MCRV .GT. 0) GO TO 181
      LINE = LINE+1
      IF (LINE .LT. 57) GO TO 180
      LINE = 8
      WRITE (XOTFIL,2000) L,DX
  180 WRITE (XOTFIL,2001) IB,XO,YO,UU,VV,XD,X1,DT,T,HMI
  181 CONTINUE
      HMI = HM
      XP = (XO-X(1))*SCLX
      YP = (YO-Y(1))*SCLY
      CALL PLOT(XP,YP,2)
      IF (MOD(IB,5) .NE. 0) GO TO 190
      S1 = HXM/X1
      S2 = -S1*UU*SCLY
      S1 = S1*VV*SCLX
      CALL PLOT(XP-S1,YP-S2,2)
      CALL PLOT(XP+S1,YP+S2,2)
      CALL PLOT(XP,YP,2)
  190 CONTINUE
      U1 = U1+UU
      V1 = V1+VV
```

```
      U2 = U2+ABS(UU)                                                   H0822900
      V2 = V2+ABS(VV)                                                   H0823000
      S1 = U1*U1+V1*V1                                                  H0823100
      S2 = U2*U2+V2*V2                                                  H0823200
      IF (S2/S1 .GE. 49.0) GO TO 200                                    H0823300
  195 CONTINUE                                                          H0823400
      IB = IB+1                                                         H0823500
      IF (IB .LT. 1000) GO TO 60                                        H0823600
  200 CONTINUE                                                          H0823700
      RETURN                                                            H0823800
 2000 FORMAT ('1',41X,'TRAJECTORY NUMBER ',I2,', DELTA-X =',F9.3,' METERH0823900
     1S'//'  POINT',5X,'X',11X,'Y',9X,'U',7X,'V',5X,'DISTANCE VELOCITY  H0824000
     2DELTA-T    TIME      HM'/' NUMBER  (METERS)    (METERS)  (M/SEC) (H0824100
     3M/SEC)  (METERS)  (M/SEC)   (SEC)    (SEC)   (METERS)'/1X,46('--')H0824200
     4)                                                                 H0824300
 2001 FORMAT (2X,I4,2F12.2,2F8.2,F10.2,2F9.2,F10.2,F9.2)                H0824400
      END                                                               H0824500
```

```
      SUBROUTINE HEC009(X,Y,ZZ,IDIM,JDIM,CONT,NCONT,SCL,LFL,NSW,Z,IFL)  H0900100
C     THIS SUBROUTINE DRAWS ALL CONTOUR PLOTS                           H0900200
      COMMON /M1/ S(124)                                                H0900300
      COMMON /DTA1/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,RAD,MAXLLH0900400
     1,XM,YM,MFLGS,SCLX,SCLY,XNCH,YNCH,J14,J15,J16,ISW7,IPLOT,MCORE     H0900500
      DIMENSION X(IDIM),Y(JDIM),ZZ(IDIM,1),Z(11,11),IFL(10,10),U(11),   H0900600
     1V(11),ZA(4,2),ZB(5),ZAB(2,3),ZX(2),ZY(2),ZXY(2),CONT(NCONT),      H0900700
     2LFL(IDIM,NCONT)                                                   H0900800
      EQUIVALENCE (J1,LXM2),(J2,LYM2),(J3,XMID),(J4,YMID),(J5,XD),      H0900900
     1(J6,YD),(J7,NSUBY),(J10,DV1),(J11,DV),(J12,NSUBY1),(J13,YJJ1),    H0901000
     2(J14,IYM2),(J15,IYM3),(J16,IYML)                                  H0901100
      EQUIVALENCE (S(1),V),(S(12),U),(S(23),ZA),(S(31),ZB),(S(36),ZAB), H0901200
     1(S(42),ZX),(S(44),ZY),(S(46),ZXY),(S(48),P00),(S(49),P01),        H0901300
     2(S(50),P10),(S(51),P11),(S(52),SW),(S(53),W2),(S(54),W3),         H0901400
     3(S(55),WX2),(S(56),WX3),(S(57),IYML1),(S(58),IX6),(S(59),NSUBX),  H0901500
     4(S(60),DU1),(S(61),DU),(S(62),NSUBX1),(S(63),XII1),(S(64),IXM1),  H0901600
     5(S(65),IXML),(S(66),Y3),(S(67),Y4),(S(68),B3),(S(69),B3SQ),       H0901700
     6(S(70),X3),(S(71),Z3B3),(S(72),A3),(S(73),A3SQ),(S(74),X4),       H0901800
     7(S(75),Z43),(S(76),A4),(S(77),X5),(S(78),Z53),(S(79),Z54),        H0901900
     8(S(80),Z5B1),(S(81),Z5B2),(S(82),Z5B3),(S(83),Z5B4),(S(84),Z5B5), H0902000
     9(S(85),X6),(S(86),Z63),(S(87),Z64),(S(88),Z6B3),(S(89),Z62)       H0902100
      EQUIVALENCE (S(90),Z6B2),(S(91),Z6B4),(S(92),Z65),(S(93),Z6B1),   H0902200
     1(S(94),Z6B5),(S(95),A5),(S(96),ZA5B2),(S(97),ZA5B3),(S(98),ZA5B4),H0902300
     2(S(99),ZX34),(S(100),ZY34),(S(101),ZXY34),(S(102),SWI),           H0902400
     3(S(103),ZX3B3),(S(104),ZX4B3),(S(105),ZY3A3),(S(106),ZY4A3),      H0902500
     4(S(107),P30),(S(108),P31),(S(109),P32),(S(110),P33),(S(111),DY),  H0902600
     5(S(112),DX),(S(113),I2),(S(114),J),(S(115),I),(S(116),I1),        H0902700
     6(S(117),KFL),(S(118),JFL),(S(119),X2),(S(120),Y2),(S(121),X1),    H0902800
     7(S(122),Y1),(S(123),TEMP),(S(124),FPN)                            H0902900
      EQUIVALENCE (Z3A2,ZA(1)),(Z3A3,ZA(2)),(Z3A4,ZA(3)),(Z3A5,ZA(4)),  H0903000
     1(Z4A2,ZA(5)),(Z4A3,ZA(6)),(Z4A4,ZA(7)),(Z4A5,ZA(8)),(Z4B1,ZB(1)), H0903100
     2(Z4B2,ZB(2)),(Z4B3,ZB(3)),(Z4B4,ZB(4)),(Z4B5,ZB(5)),(ZA3B2,ZAB(1))H0903200
     3,(ZA4B2,ZAB(2)),(ZA3B3,ZAB(3)),(ZA4B3,ZAB(4)),(ZA3B4,ZAB(5)),     H0903300
     4(ZA4B4,ZAB(6)),(ZX43,ZX(1)),(ZX44,ZX(2)),(ZY43,ZY(1)),(ZY44,ZY(2))H0903400
     5,(ZXY43,ZXY(1)),(ZXY44,ZXY(2)),(P00,Z33),(P01,ZY33),(P10 ,ZX33),  H0903500
     6(P11,ZXY33)                                                       H0903600
      EQUIVALENCE (SW,E),(W2,WY2,A,Q0),(W3,WY3,B,Q1),(WX2,C,Q2),(WX3,D, H0903700
     1Q3),(Z3A2,P02),(Z4A2,P03),(Z4B1,P12),(Z4B2,P13),(Z4B4,P20),(Z4B5, H0903800
```

```
     2P21),(ZA3B2,P22),(ZA3B4,P23)                                      H0903900
      LXM2 = IDIM-2                                                     H0904000
      LYM2 = JDIM-2                                                     H0904100
      DO 40 J=1,NCONT                                                   H0904200
      DO 40 I=1,IDIM                                                    H0904300
   40 LFL(I,J) = 0                                                      H0904400
      XMID = 0.5*(X(IDIM)+X(1))                                         H0904500
      YMID = 0.5*(Y(JDIM)+Y(1))                                         H0904600
      XD = (X(IDIM)-X(1))*SCL*0.5                                       H0904700
      YD = (Y(JDIM)-Y(1))*SCL*0.5                                       H0904800
C                                                                       H0904900
C         MAIN DO LOOP OVER INPUT DATA GRID                             H0905000
C                                                                       H0905100
      DO 1260 JJ=2,JDIM                                                 H0905200
      NSUBY = 6                                                         H0905300
      DV1 = (Y(JJ)-Y(JJ-1))*0.2                                         H0905400
      DV = DV1*SCL                                                      H0905500
      IF (DV .LE. 0.05) GO TO 70                                        H0905600
      NSUBY = DV*100.0+1.5                                              H0905700
      IF (NSUBY .GT. 11) NSUBY = 11                                     H0905800
      DV1 = (Y(JJ)-Y(JJ-1))/(NSUBY-1)                                   H0905900
      DV = DV1*SCL                                                      H0906000
   70 NSUBY1 = NSUBY-1                                                  H0906100
      YJJ1 = (Y(JJ-1)-YMID)*SCL                                         H0906200
      DO 110 J=1,NSUBY                                                  H0906300
  110 V(J) = YJJ1+(J-1)*DV                                              H0906400
      IYM2 = JJ-2                                                       H0906500
      IYM3 = IYM2-1                                                     H0906600
      IYML = JJ-JDIM                                                    H0906700
      IYML1 = IYML+1                                                    H0906800
      IX6 = 0                                                           H0906900
C                                                                       H0907000
C         MAIN DO LOOP OVER INPUT DATA GRID                             H0907100
C                                                                       H0907200
      DO 1250 II=1,IDIM                                                 H0907300
      IF (II .EQ. 1) GO TO 130                                          H0907400
      NSUBX = 6                                                         H0907500
      DU1 = (X(II)-X(II-1))*0.2                                         H0907600
```

```
      DU = DU1*SCL                                                     H0907700
      IF (DU .LE. 0.05) GO TO 117                                      H0907800
      NSUBX = DU*100.0+1.5                                             H0907900
      IF (NSUBX .GT. 11) NSUBX = 11                                    H0908000
      DU1 = (X(II)-X(II-1))/(NSUBX-1)                                  H0908100
      DU = DU1*SCL                                                     H0908200
  117 NSUBX1 = NSUBX-1                                                 H0908300
      XII1 = (X(II-1)-XMID)*SCL                                        H0908400
      CALL PLOT(XII1+XD,YJJ1+YD,3)                                     H0908500
      CALL PLOT(XII1+XD,YJJ1+YD,2)                                     H0908600
      DO 120 I=1,NSUBX                                                 H0908700
  120 U(I) = XII1+(I-1)*DU                                             H0908800
  130 CONTINUE                                                         H0908900
      IXM1 = II-1                                                      H0909000
      IXML = II-IDIM                                                   H0909100
      IF (IXM1 .NE. 0) GO TO 150                                       H0909200
      Y3 = Y(JJ-1)                                                     H0909300
      Y4 = Y(JJ)                                                       H0909400
      B3 = 1.0/(Y4-Y3)                                                 H0909500
      B3SQ = B3*B3                                                     H0909600
      IF (IYM2 .GT. 0) B2 = 1.0/(Y3-Y(IYM2))                           H0909700
      IF (IYM3 .GT. 0) B1 = 1.0/(Y(IYM2)-Y(IYM3))                      H0909800
      IF (IYML .LT. 0) B4 = 1.0/(Y(JJ+1)-Y4)                           H0909900
      IF (IYML1 .LT. 0) B5 = 1.0/(Y(JJ+2)-Y(JJ+1))                     H0910000
      GO TO 180                                                        H0910100
  150 Z3A2 = Z3A3                                                      H0910200
      Z4A2 = Z4A3                                                      H0910300
      X3 = X4                                                          H0910400
      Z33 = Z43                                                        H0910500
      Z3B3 = Z4B3                                                      H0910600
      A3 = A4                                                          H0910700
      A3SQ = A3*A3                                                     H0910800
      Z3A3 = Z3A4                                                      H0910900
      Z4A3 = Z4A4                                                      H0911000
      ZA3B2 = ZA4B2                                                    H0911100
      ZA3B3 = ZA4B3                                                    H0911200
      ZA3B4 = ZA4B4                                                    H0911300
  160 X4 = X5                                                          H0911400
```

```
      Z43 = Z53                                      H0911500
      Z4B1 = Z5B1                                    H0911600
      Z4B2 = Z5B2                                    H0911700
      Z4B3 = Z5B3                                    H0911800
      Z4B4 = Z5B4                                    H0911900
      Z4B5 = Z5B5                                    H0912000
      A4 = A5                                        H0912100
      Z3A4 = Z3A5                                    H0912200
      Z4A4 = Z4A5                                    H0912300
      ZA4B2 = ZA5B2                                  H0912400
      ZA4B3 = ZA5B3                                  H0912500
      ZA4B4 = ZA5B4                                  H0912600
  170 X5 = X6                                        H0912700
      Z53 = Z63                                      H0912800
      Z54 = Z64                                      H0912900
      Z5B1 = Z6B1                                    H0913000
      Z5B2 = Z6B2                                    H0913100
      Z5B3 = Z6B3                                    H0913200
      Z5B4 = Z6B4                                    H0913300
      Z5B5 = Z6B5                                    H0913400
  180 IX6 = IX6+1                                    H0913500
      IF (IX6 .GT. IDIM) GO TO 260                   H0913600
      X6 = X(IX6)                                    H0913700
      Z63 = ZZ(IX6,JJ-1)                             H0913800
      Z64 = ZZ(IX6,JJ)                               H0913900
      Z6B3 = (Z64-Z63)*B3                            H0914000
      IF (LYM2 .EQ. 0) GO TO 200                     H0914100
      IF (IYM2 .EQ. 0) GO TO 190                     H0914200
      Z62 =ZZ(IX6,JJ-2)                              H0914300
      Z6B2 = (Z63-Z62)*B2                            H0914400
      IF (IYML .NE. 0) GO TO 190                     H0914500
      Z6B4 = Z6B3+Z6B3-Z6B2                          H0914600
      GO TO 210                                      H0914700
  190 Z65 =ZZ(IX6,JJ+1)                              H0914800
      Z6B4 = (Z65-Z64)*B4                            H0914900
      IF (IYM2 .NE. 0) GO TO 210                     H0915000
      Z6B2 = Z6B3+Z6B3-Z6B4                          H0915100
      GO TO 210                                      H0915200
```

```
  200 Z6B2 = Z6B3                                          H0915300
      Z6B4 = Z6B3                                          H0915400
  210 IF (IYM3 .LE. 0) GO TO 220                           H0915500
      Z6B1 = (Z62-ZZ(IX6,JJ-3))*B1                         H0915600
      GO TO 230                                            H0915700
  220 Z6B1 = Z6B2+Z6B2-Z6B3                                H0915800
  230 IF (IYML1 .GE. 0) GO TO 240                          H0915900
      Z6B5 = (ZZ(IX6,JJ+2)-Z65)*B5                         H0916000
      GO TO 250                                            H0916100
  240 Z6B5 = Z6B4+Z6B4-Z6B3                                H0916200
  250 IF (IX6 .EQ. 1) GO TO 170                            H0916300
      A5 = 1.0/(X6-X5)                                     H0916400
      Z3A5 = (Z63-Z53)*A5                                  H0916500
      Z4A5 = (Z64-Z54)*A5                                  H0916600
      ZA5B2 = (Z6B2-Z5B2)*A5                               H0916700
      ZA5B3 = (Z6B3-Z5B3)*A5                               H0916800
      ZA5B4 = (Z6B4-Z5B4)*A5                               H0916900
      IF (IX6 .EQ. 2) GO TO 160                            H0917000
      GO TO 280                                            H0917100
  260 IF (LXM2 .EQ. 0) GO TO 270                           H0917200
      Z3A5 = Z3A4+Z3A4-Z3A3                                H0917300
      Z4A5 = Z4A4+Z4A4-Z4A3                                H0917400
      IF (IXML .EQ. 0) GO TO 290                           H0917500
      ZA5B2 = ZA4B2+ZA4B2-ZA3B2                            H0917600
      ZA5B3 = ZA4B3+ZA4B3-ZA3B3                            H0917700
      ZA5B4 = ZA4B4+ZA4B4-ZA3B4                            H0917800
      GO TO 290                                            H0917900
  270 Z3A5 = Z3A4                                          H0918000
      Z4A5 = Z4A4                                          H0918100
      IF (IXML .EQ. 0) GO TO 290                           H0918200
      ZA5B2 = ZA4B2                                        H0918300
      ZA5B3 = ZA4B3                                        H0918400
      ZA5B4 = ZA4B4                                        H0918500
  280 IF (IXM1 .NE. 0) GO TO 290                           H0918600
      Z3A3 = Z3A4+Z3A4-Z3A5                                H0918700
      Z3A2 = Z3A3+Z3A3-Z3A4                                H0918800
      Z4A3 = Z4A4+Z4A4-Z4A5                                H0918900
      Z4A2 = Z4A3+Z4A3-Z4A4                                H0919000
```

```
      ZA3B2 = ZA4B2+ZA4B2-ZA5B2                                         H0919100
      ZA3B3 = ZA4B3+ZA4B3-ZA5B3                                         H0919200
      ZA3B4 = ZA4B4+ZA4B4-ZA5B4                                         H0919300
      GO TO 300                                                         H0919400
  290 ZX33 = ZX43                                                       H0919500
      ZX34 = ZX44                                                       H0919600
      ZY33 = ZY43                                                       H0919700
      ZY34 = ZY44                                                       H0919800
      ZXY33 = ZXY43                                                     H0919900
      ZXY34 = ZXY44                                                     H0920000
  300 DO 350 JY=1,2                                                     H0920100
      W2 = ABS(ZA(4,JY)-ZA(3,JY))                                       H0920200
      W3 = ABS(ZA(2,JY)-ZA(1,JY))                                       H0920300
      SW = W2+W3                                                        H0920400
      IF (SW) 310,310,305                                               H0920500
  305 SWI = 1.0/SW                                                      H0920600
      WX2 = W2*SWI                                                      H0920700
      WX3 = W3*SWI                                                      H0920800
      GO TO 320                                                         H0920900
  310 WX2 = 0.5                                                         H0921000
      WX3 = 0.5                                                         H0921100
  320 ZX(JY) = WX2*ZA(2,JY)+WX3*ZA(3,JY)                                H0921200
      W2 = ABS(ZB(JY+3)-ZB(JY+2))                                       H0921300
      W3 = ABS(ZB(JY+1)-ZB(JY))                                         H0921400
      SW = W2+W3                                                        H0921500
      IF (SW) 330,330,325                                               H0921600
  325 SWI = 1.0/SW                                                      H0921700
      WY2 = W2*SWI                                                      H0921800
      WY3 = W3*SWI                                                      H0921900
      GO TO 340                                                         H0922000
  330 WY2 = 0.5                                                         H0922100
      WY3 = 0.5                                                         H0922200
  340 ZY(JY) = WY2*ZB(JY+1)+WY3*ZB(JY+2)                                H0922300
      ZXY(JY) = WY2*(WX2*ZAB(1,JY)+WX3*ZAB(2,JY))+WY3*(WX2*ZAB(1,JY+1)+ H0922400
     1WX3*ZAB(2,JY+1))                                                  H0922500
  350 CONTINUE                                                          H0922600
      IF (IXM1 .EQ. 0) GO TO 1250                                       H0922700
      ZX3B3 = (ZX34-ZX33)*B3                                            H0922800
```

```
      ZX4B3 = (ZX44-ZX43)*B3
      ZY3A3 = (ZY43-ZY33)*A3
      ZY4A3 = (ZY44-ZY34)*A3
      A = ZA3B3-ZX3B3-ZY3A3+ZXY33
      B = ZX4B3-ZX3B3-ZXY43+ZXY33
      C = ZY4A3-ZY3A3-ZXY34+ZXY33
      D = ZXY44-ZXY43-ZXY34+ZXY33
      E = A+A-B-C
      P02 = (2.0*(Z3B3-ZY33)+Z3B3-ZY34)*B3
      P03 = (-2.0*Z3B3+ZY34+ZY33)*B3SQ
      P12 = (2.0*(ZX3B3-ZXY33)+ZX3B3-ZXY34)*B3
      P13 = (-2.0*ZX3B3+ZXY34+ZXY33)*B3SQ
      P20 = (2.0*(Z3A3-ZX33)+Z3A3-ZX43)*A3
      P21 = (2.0*(ZY3A3-ZXY33)+ZY3A3-ZXY43)*A3
      P22 = (3.0*(A+E)+D)*A3*B3
      P23 = (-3.0*E-B-D)*A3*B3SQ
      P30 = (-2.0*Z3A3+ZX43+ZX33)*A3SQ
      P31 = (-2.0*ZY3A3+ZXY43+ZXY33)*A3SQ
      P32 = (-3.0*E-C-D)*B3*A3SQ
      P33 = (D+E+E)*A3SQ*B3SQ
      DO 370 JY=1,NSUBY
      DY = (JY-1)*DV1
      Q0 = P00+DY*(P01+DY*(P02+DY*P03))
      Q1 = P10+DY*(P11+DY*(P12+DY*P13))
      Q2 = P20+DY*(P21+DY*(P22+DY*P23))
      Q3 = P30+DY*(P31+DY*(P32+DY*P33))
      DO 360 JX=1,NSUBX
      DX = (JX-1)*DU1
      Z(JX,JY) = Q0+DX*(Q1+DX*(Q2+DX*Q3))
  360 CONTINUE
  370 CONTINUE
  380 CONTINUE
C
C          DO LOOP OVER CONTOUR (ISOPLETH) VALUES
C
      DO 1240 K=1,NCONT
      DO 390 J=1,NSUBY1
      DO 390 I=1,NSUBX1
```

```
  390 IFL(I,J) = 0                                               H0926700
      C = CONT(K)                                                H0926800
      I2 = 0                                                     H0926900
C                                                                H0927000
C        LOOPS OVER SUBDIVILED GRID CELL                         H0927100
C                                                                H0927200
      DO 1200 JA=1,NSUBY1                                        H0927300
      DO 1190 IA=1,NSUBX1                                        H0927400
      J = JA                                                     H0927500
      I = IA                                                     H0927600
      I1 = 0                                                     H0927700
      IF (IFL(I,J) .NE. 0) GO TO 1190                            H0927800
      KFL = 0                                                    H0927900
      JFL = 0                                                    H0928000
 1040 IF ((Z(I,J)-C)*(Z(I+1,J)-C) .GT. 0.0) GO TO 1050           H0928100
      X2 = Z(I,J)-Z(I+1,J)                                       H0928200
      IF (X2) 1042,1041,1042                                     H0928300
 1041 X2 = 1.0                                                   H0928400
      GO TO 1043                                                 H0928500
 1042 X2 = (Z(I,J)-C)/X2                                         H0928600
 1043 X2 = U(I)+DU*X2                                            H0928700
      Y2 = V(J)                                                  H0928800
      IF (I1 .NE. 0) GO TO 1115                                  H0928900
      X1 = X2                                                    H0929000
      Y1 = Y2                                                    H0929100
      I1 = 1                                                     H0929200
 1050 IF ((Z(I,J)-C)*(Z(I,J+1)-C) .GT. 0.0) GO TO 1060           H0929300
      X2 = U(I)                                                  H0929400
      Y2 = Z(I,J)-Z(I,J+1)                                       H0929500
      IF (Y2) 1052,1051,1052                                     H0929600
 1051 Y2 = 1.0                                                   H0929700
      GO TO 1053                                                 H0929800
 1052 Y2 = (Z(I,J)-C)/Y2                                         H0929900
 1053 Y2 = V(J)+DV*Y2                                            H0930000
      IF (I1 .NE. 0) GO TO 1120                                  H0930100
      X1 = X2                                                    H0930200
      Y1 = Y2                                                    H0930300
      I1 = 2                                                     H0930400
```

```
 1060 IF ((Z(I,J+1)-C)*(Z(I+1,J+1)-C) .GT. 0.0) GO TO 1070                H0930500
      X2 = Z(I,J+1)-Z(I+1,J+1)                                             H0930600
      IF (X2) 1062,1061,1062                                               H0930700
 1061 X2 = 1.0                                                             H0930800
      GO TO 1063                                                           H0930900
 1062 X2 = (Z(I,J+1)-C)/X2                                                 H0931000
 1063 X2 = U(I)+DU*X2                                                      H0931100
      Y2 = V(J+1)                                                          H0931200
      IF (I1 .NE. 0) GO TO 1130                                            H0931300
      X1 = X2                                                              H0931400
      Y1 = Y2                                                              H0931500
      I1 = 3                                                               H0931600
 1070 IF (I1 .EQ. 0) GO TO 1190                                            H0931700
 1080 IF ((Z(I+1,J)-C)*(Z(I+1,J+1)-C) .GT. 0.0) GO TO 1040                H0931800
      X2 = U(I+1)                                                          H0931900
      Y2 = Z(I+1,J)-Z(I+1,J+1)                                             H0932000
      IF (Y2) 1082,1081,1082                                               H0932100
 1081 Y2 = 1.0                                                             H0932200
      GO TO 1083                                                           H0932300
 1082 Y2 = (Z(I+1,J)-C)/Y2                                                 H0932400
 1083 Y2 = V(J)+DV*Y2                                                      H0932500
      GO TO 1140                                                           H0932600
 1115 IFL(I,J) = 1                                                         H0932700
 1117 IF (J .EQ. 1) GO TO 1150                                             H0932800
      IF (IFL(I,J-1) .NE. 0) GO TO 1150                                    H0932900
      J = J-1                                                              H0933000
      IF (KFL .NE. 0) GO TO 1118                                           H0933100
      KFL = 1                                                              H0933200
      CALL PLOT(X1+XD,Y1+YD,3)                                             H0933300
 1118 CALL PLOT(X2+XD,Y2+YD,2)                                             H0933400
      I1 = 3                                                               H0933500
      JFL = 0                                                              H0933600
      X1 = X2                                                              H0933700
      Y1 = Y2                                                              H0933800
      GO TO 1080                                                           H0933900
 1120 IFL(I,J) = 1                                                         H0934000
 1125 IF (I .EQ. 1) GO TO 1150                                             H0934100
      IF (IFL(I-1,J) .NE. 0) GO TO 1150                                    H0934200
```

```
      I = I-1                                                           H0934300
      IF (KFL .NE. 0) GO TO 1127                                        H0934400
      KFL = 1                                                           H0934500
      CALL PLOT(X1+XD,Y1+YD,3)                                          H0934600
 1127 CALL PLOT(X2+XD,Y2+YD,2)                                          H0934700
      I1 = 4                                                            H0934800
      JFL = 0                                                           H0934900
      X1 = X2                                                           H0935000
      Y1 = Y2                                                           H0935100
      GO TO 1040                                                        H0935200
 1130 IFL(I,J) = 1                                                      H0935300
 1135 IF (J .EQ. NSUBY1) GO TO 1150                                     H0935400
      IF (IFL(I,J+1) .NE. 0) GO TO 1150                                 H0935500
      J = J+1                                                           H0935600
      IF (KFL .NE. 0) GO TO 1137                                        H0935700
      KFL = 1                                                           H0935800
      CALL PLOT(X1+XD,Y1+YD,3)                                          H0935900
 1137 CALL PLOT(X2+XD,Y2+YD,2)                                          H0936000
      I1 = 1                                                            H0936100
      JFL = 0                                                           H0936200
      X1 = X2                                                           H0936300
      Y1 = Y2                                                           H0936400
      GO TO 1050                                                        H0936500
 1140 IFL(I,J) = 1                                                      H0936600
 1145 IF (I .EQ. NSUBX1) GO TO 1150                                     H0936700
      IF (IFL(I+1,J) .NE. 0) GO TO 1150                                 H0936800
      I = I+1                                                           H0936900
      IF (KFL .NE. 0) GO TO 1147                                        H0937000
      KFL = 1                                                           H0937100
      CALL PLOT(X1+XD,Y1+YD,3)                                          H0937200
 1147 CALL PLOT(X2+XD,Y2+YD,2)                                          H0937300
      I1 = 2                                                            H0937400
      JFL = 0                                                           H0937500
      X1 = X2                                                           H0937600
      Y1 = Y2                                                           H0937700
      GO TO 1060                                                        H0937800
 1150 IF (JFL .NE. 0) GO TO 1160                                        H0937900
      JFL = 1                                                           H0938000
```

```
      TEMP = X1                                                         H0938100
      X1 = X2                                                           H0938200
      X2 = TEMP                                                         H0938300
      TEMP = Y1                                                         H0938400
      Y1 = Y2                                                           H0938500
      Y2 = TEMP                                                         H0938600
      GO TO (1117,1125,1135,1145),I1                                    H0938700
 1160 IF (KFL .EQ. 0) CALL PLOT(X2+XD,Y2+YD,3)                          H0938800
      CALL PLOT(X1+XD,Y1+YD,2)                                          H0938900
      I2 = 1                                                            H0939000
 1190 CONTINUE                                                          H0939100
 1200 CONTINUE                                                          H0939200
      IF (LFL(II,K)+LFL(II-1,K) .NE. 0) GO TO 1230                      H0939300
      IF (I2 .EQ. 0) GO TO 1230                                         H0939400
      FPN = K                                                           H0939500
      CALL NUMBER(X2+XD,Y2+YD,0.05,FPN,0.0,-1)                          H0939600
 1230 LFL(II,K) = I2                                                    H0939700
 1240 CONTINUE                                                          H0939800
 1250 CONTINUE                                                          H0939900
 1260 CONTINUE                                                          H0940000
      Y1 = -0.9                                                         H0940100
      Y2 = 0.0                                                          H0940200
      IF (NSW .NE. 0) GO TO 1300                                        H0940300
      X1 = 0.5*(XD-1.8)                                                 H0940400
      CALL HEC014(X1,Y1,.075,24HTERRAIN ELEVATION LEVELS,0.0,24)        H0940500
      GO TO 1321                                                        H0940600
 1300 IF (NSW .LT. 0) GO TO 1310                                        H0940700
      X1 = 0.5*(XD-1.35)+XD                                             H0940800
      CALL HEC014(X1,Y1,.075,18HLAYER DEPTH LEVELS,0.0,18)              H0940900
      GO TO 1320                                                        H0941000
 1310 X1 = 0.5*(XD-1.65)+XD                                             H0941100
      CALL HEC014(X1,Y1,.075,22HLAYER ELEVATION LEVELS,0.0,22)          H0941200
 1320 Y2 = XD                                                           H0941300
 1321 CONTINUE                                                          H0941400
      Y1 = -1.0                                                         H0941500
      X1 = 0.5*(XD-2.7)-0.9+Y2                                          H0941600
      J = 0                                                             H0941700
      I = 0                                                             H0941800
```

```
 1330 J = J+1                                                           H0941900
      IF (J .GT. NCONT) GO TO 1350                                      H0942000
      I = I+1                                                           H0942100
      X1 = X1+0.9                                                       H0942200
      IF (I .LT. 4) GO TO 1340                                          H0942300
      I = 1                                                             H0942400
      X1 = 0.5*(XD-2.7)+Y2                                              H0942500
      Y1 = Y1-0.1                                                       H0942600
 1340 X2 = J                                                            H0942700
      CALL NUMBER(X1,Y1,.075,X2,0.0,-1)                                 H0942800
      CALL HEC014(X1+0.15,Y1,.075,1H=,0.0,1)                            H0942900
      CALL NUMBER(X1+0.225,Y1,.075,CONT(J),0.0,1)                       H0943000
      CALL HEC014(X1+0.75,Y1,.075,2H, ,0.0,2)                           H0943100
      GO TO 1330                                                        H0943200
 1350 CONTINUE                                                          H0943300
      RETURN                                                            H0943400
      END                                                               H0943500
```

```
      SUBROUTINE HEC013(XM,YM,IFST)                                     H1300100
C*C*C THIS SUBROUTINE IS THE WSMR VERSION, LOOK AT /UOFU FOR U OF U     H1300200
C*C*C VERSION - DO NOT INCLUDE IN U OF U COLLECTION                     H1300300
      IF (IFST .NE. 0) GO TO 10                                         H1300400
      IFST = 1                                                          H1300500
      CALL INITAL(1,400,36,0,0,0)                                       H1300600
      CALL HEC014(-1.0,4.0,0.2,24HSTART PLOTS FOR OHMSTEDE,90.0,24)     H1300700
      CALL HEC014(-0.5,3.0,0.2,32HROOM 203, BLDG 1622, PH 678-3123,90.0,H1300800
     132)                                                               H1300900
   10 CALL PLOT(1.5,3.0,-3)                                             H1301000
      CALL HEC016(0)                                                    H1301100
      RETURN                                                            H1301200
      END                                                               H1301300
```

```
      SUBROUTINE HEC014(X,Y,H,A,T,N)                                   H1400100
C*C*C THIS SUBROUTINE IS THE WSMR VERSION, LOOK AT /UOFU FOR U OF U    H1400200
C*C*C VERSION - DO NOT INCLUDE IN U OF U COLLECTION                    H1400300
      CALL SYMBOL(X,Y,H,A,T,N)                                         H1400400
      RETURN                                                           H1400500
      END                                                              H1400600
```

```
      SUBROUTINE HEC015(JSW,XM)                                         H1500100
C*C*C THIS SUBROUTINE IS THE WSMR VERSION, LOOK AT /UOFU FOR U OF U     H1500200
C*C*C VERSION - DO NOT INCLUDE IN U OF U COLLECTION                     H1500300
      CALL RSTR(2)                                                      H1500400
      CALL PLOT(1.5,3.0,-3)                                             H1500500
      IF (JSW .EQ. 0) GO TO 10                                          H1500600
      CALL HEC014(0.0,3.0,0.3,12HEND OF PLOTS,0.0,12)                   H1500700
      CALL RSTR(3)                                                      H1500800
   10 CONTINUE                                                          H1500900
      RETURN                                                            H1501000
      END                                                               H1501100
```

```
      SUBROUTINE HEC016(N)                                              H1600100
C*C*C THIS SUBROUTINE IS THE WSMR VERSION, LOOK AT /UOFU FOR U OF U     H1600200
C*C*C VERSION - DO NOT INCLUDE IN U OF U COLLECTION                     H1600300
      CALL NEWPN(N+1)                                                   H1600400
      RETURN                                                            H1600500
      END                                                               H1600600
```

```
. SUBROUTINE DRMRD/WT
. DRUM MASS STORAGE HANDLER CALLABLE BY FORTRAN
. CALL SEQUENCE  CALL DRMRD/WT(N,IARRY,MPA,MPL,MAD,FN)
.       N=NO. OF 36 BIT WORDS TO TRANSFERE
.       IARRAY STARTING ADDRESS OF DATA
.       STARTING DRUM ADDRESS=MPA*MPL+MAD
.       FN=DRUM FILE NAME AS ON  USE CARD
. RESTRICTIONS
.       MAX WORDS TRANSFERED 2**16-1
.       MAX DRUM ADDRESS 2**24-1
.       MAX FN 99999
. COMMENTS
.       DRUM ADDRESSING STARTS AT ZERO WHILE FORTRAN STARTS AT ONE
. SUBROUTINE FSTRD/WT
. FASTRAN MASS STORAGE HANDLER CALLABLE BY FORTRAN
. FOR CALLING SEQUENCE AND RESTRICTIONS SEE DRMRD/WT AND SUBSTITUTE 'DISK' FOR
. 'DRUM'.
. ADDITIONAL COMMENTS
.       THE SMALLEST ADDRESSABLE GRANULE IS 28 36 BIT WORDS
.       THEREFORE THE  DISK ADDRESS MPA*MPL+MAD MUST BE A MULIPLE OF 28
.       DISK ADDRESS ZERO REFERS TO THE FIRST 28 WORDS
.       THIS MOD(28) FEATURE MUST BE HANDLED BY THE USER
. BY DCP
. START 8/28/74
. VERSION 1.01 9/3/74
            AXR$
FSTWT*      L      A0,WRITE
            J      $+2
FSTRD*      L      A0,READ
            L      A1,F
            L      A2,(+'FST')
            J      FUNCT
DRMWT*      L      A0,WRITE
            J      $+2
DRMRD*      L      A0,READ
            L      A1,D
            L      A2,(+'DRM')
FUNCT       S      A1,DEVICE
```

```
         S,H2   A2,NTOBIG+1
         S,H2   A2,DTOBIG+1
         S,H2   A2,FNNOGD+1
         S,H1   A2,LINE1
         S      A0,IO+3           . PLACE READ/WRITE FUNCTION IN I/O PACKET
         SZ     IO+2
         L      A0,TYPE
         S      A0,ERROR
         L      A0,*0,X11         . N
         TNG    A0,(1*/16)        . TEST N .GT. 2**16-1
         J      NOK
         L      A0,(PF 5,6,NTOBIG)
         J      PRINT
NOK      S,H1   A0,IO+4           . PLACE WORD COUNT N IN I/O PACKET
         L      A0,1,X11          . IARRY
         S,H2   A0,IO+4           . PLACE STARTING ADDRESS-IARRY IN I/O PACKET
         L      A0,*2,X11         .MPA
         MI     A0,*3,X11         . MPA*MPL
         A      A1,*4,X11         . MPA*MPL+MAD
         JNZ    A0,$+3            . TEST DRUM ADDRESS .GT. 2**24-1
         TNG    A1,(1*/24)
         J      DOK
         L      A0,(PF 5,8,DTOBIG)
         J      PRINT
DOK      S      A1,IO+5           . PLACE DRUM ADDRESS IN I/O PACKET
         L      A2,BLANK
         S      A2,IO+1
         L,U    A0,0
         L      A1,(99999)        . TEST FN WITHIN RANGE
         TW     A0,*5,X11         . 0 .LE. FN .LE. 99999
         TNZ    *5,X11
         J      FNOK
         L      A0,(PF 5,6,FNNOGD)
         J      PRINT
FNOK     LM     A1,*5,X11
AGAIN    DI,U   A0,10             . CONVERT FN TO DECIMAL
         A,U    A1,060            . CONVERT DECIMAL TO FIELD DATA
         DSL    A1,6              . STORE IN A2
```

```
               JZ      A0,$+3          . CHECK IF DONE
               DSL     A0,36
               J       AGAIN
               SSL     A2,6
               A       A2,DEVICE
               S       A2,IO           . THIS COMPLETES I/O PACKET WITH FILENAME
               L       A0,(PF 01000,1,ERROR)    . TELL EXEC-8 IF CONTINGENCY ERROR
               ER      IALL$           . TYPE 012 OCCOURS IT WILL BE PROCESSED AT 'ERROR'
               L,U     A0,IO
               ER      IOW$            . PERFORM I/O
               L,S1    A0,IO+3         . OBTAIN STATUS
               JZ      A0,RETURN
               L,U     A5,0            . IF PROGRAM GETS THIS FAR WE HAVE AN I/O ERROR
               TE,U    A0,022          . OTHER THAN A CONTINGENCY
               J       $+2
               L,U     A5,1            . SET EOF ERROR FLAG
               S,S2    A0,ERROR
               J       $+3
     ERROR     +0      . PROVIDE SPACE FOR CONTINGENCY PACKET
               +0
               L       A0,ERROR
               SSL     A0,18
               L,U     A2,6
               L       A1,BLANK
     TAG       DSL     A0,3            . CONVERT ERROR MESSAGE TO FIELD DATA
               SSL     A1,3            . T1(A1)=ERROR TYPE
               A       A1,FIELD        . T2(A1)=ERROR CODE
               AN,U    A2,1            . T3(A1)=CONTINGENCY TYPE
               JNZ     A2,TAG
               S,T3    A1,LINE2+3
               SSL     A1,12
               S,T3    A1,LINE4+3
               SSL     A1,12
               S,T3    A1,LINE3+3
               DL      A0,IO
               DS      A0,LINE1+4
               L       A0,(PF 5,6,LINE1)
               ER      PRINT$
```

```
                L      A0,(PF 1,4,LINE2)
                ER     PRINT$
                L      A0,(PF 1,4,LINE3)
                ER     PRINT$
                L      A0,(PF 1,4,LINE4)
                ER     PRINT$
                JZ     A5,STOP
                L      A0,(PF 1,3,EOF)
       PRINT    ER     PRINT$
       STOP     ER     EABT$
       RETURN   J      7,X11
       PF       FORM   12,6,18
       WRITE    001000000000
       READ     002000000000
       FIELD    0600000000000
       DEVICE   RES    1
       BLANK    '      '
       F        0130000000000
       D        0110000000000
       TYPE     0010012000000
       LINE1    'XXXRD/WT ERROR IN FILE
       LINE2    ' CONTINGENCY TYPE     000'
       LINE3    ' ERROR TYPE           000'
       LINE4    ' ERROR CODE           000'
       EOF      ' *END OF FILE*'
       NTOBIG   ' ERR  IN XXXRD/WT N .GT. 2**16-1'
       DTOBIG   ' ERR  IN XXXRD/WT START ADDRESS .GT. 2**24-1'
       FNNOGD   ' ERR  IN XXXRD/WT FN OUT OF RANGE'
       IO       RES    6
                END
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1977—777-022/13

DEPARTMENT OF THE ARMY
ATMOSPHERIC SCIENCES LABORATORY
US ARMY ELECTRONICS COMMAND
DRSEL-BL-DP-P
WHITE SANDS MISSILE RANGE
NEW MEXICO 88002

OFFICIAL BUSINESS
Penalty For Private Use, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314

U.S.MAIL